Blackfin®
Embedded Processor

# ADSP-BF534/ADSP-BF536/ADSP-BF537

## FEATURES

**Up to 600 MHz high performance Blackfin processor**
- **Two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs, 40-bit shifter**
- **RISC-like register and instruction model for ease of programming and compiler-friendly support**
- **Advanced debug, trace, and performance monitoring**

**0.8 V to 1.2 V core $V_{DD}$ with on-chip voltage regulation**
**2.5 V and 3.3 V-compliant I/O with specific 5 V-tolerant pins**
**182-ball and 208-ball MBGA packages**

## MEMORY

**Up to 132K bytes of on-chip memory comprised of:**
- **Instruction SRAM/cache; instruction SRAM; data SRAM/cache; additional dedicated data SRAM; scratchpad SRAM (see Table 1 on Page 3 for available memory configurations)**

**External memory controller with glueless support for SDRAM and asynchronous 8-bit and 16-bit memories**
**Flexible booting options from external flash, SPI and TWI memory or from SPI, TWI, and UART host devices**
**Memory management unit providing memory protection**

## PERIPHERALS

**IEEE 802.3-compliant 10/100 Ethernet MAC (ADSP-BF536 and ADSP-BF537 only)**
**Controller area network (CAN) 2.0B interface**
**Parallel peripheral interface (PPI), supporting ITU-R 656 video data formats**
**Two dual-channel, full-duplex synchronous serial ports (SPORTs), supporting eight stereo I²S channels**
**12 peripheral DMAs, 2 mastered by the Ethernet MAC**
**Two memory-to-memory DMAs with external request lines**
**Event handler with 32 interrupt inputs**
**Serial peripheral interface (SPI)-compatible**
**Two UARTs with IrDA® support**
**Two-wire interface (TWI) controller**
**Eight 32-bit timer/counters with PWM support**
**Real-time clock (RTC) and watchdog timer**
**32-bit core timer**
**48 general-purpose I/Os (GPIOs), 8 with high current drivers**
**On-chip PLL capable of 1× to 63× frequency multiplication**
**Debug/JTAG interface**

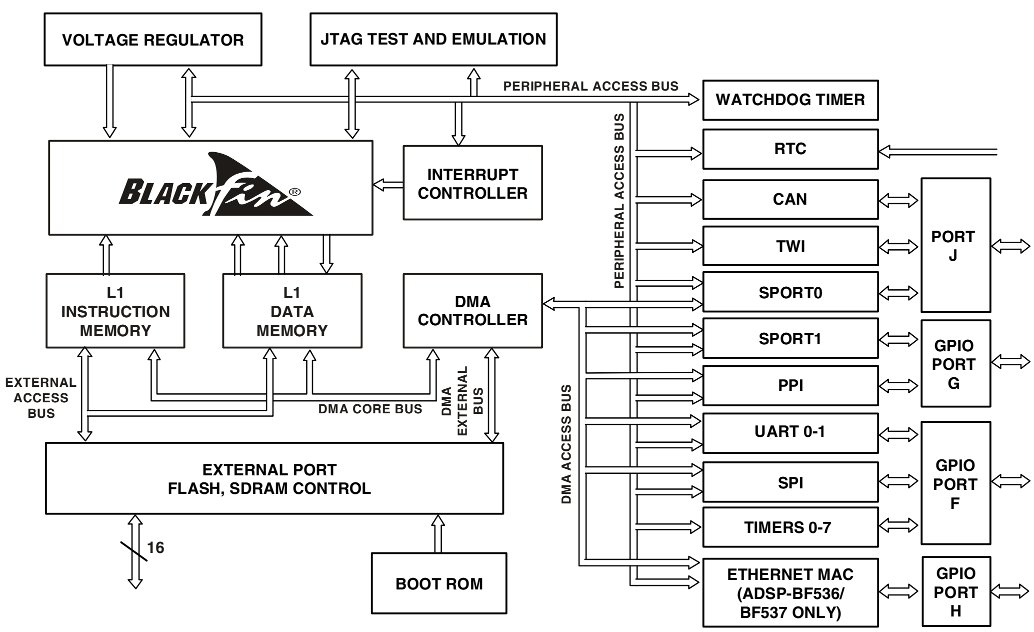



*Figure 1. Functional Block Diagram*

Blackfin and the Blackfin logo are registered trademarks of Analog Devices, Inc.

**Rev. C**




**One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106 U.S.A.**
**Tel: 781.329.4700 www.analog.com**
**Fax: 781.461.3113**

# TABLE OF CONTENTS


General Description .... 3
- Portable Low Power Architecture .... 3
- System Integration .... 3
- Blackfin Processor Peripherals .... 3
- Blackfin Processor Core .... 4
- Memory Architecture .... 5
- DMA Controllers .... 8
- Real-Time Clock .... 9
- Watchdog Timer .... 9
- Timers .... 9
- Serial Ports (SPORTs) .... 10
- Serial Peripheral Interface (SPI) Port .... 10
- UART Ports .... 10
- Controller Area Network (CAN) .... 11
- TWI Controller Interface .... 11
- 10/100 Ethernet MAC .... 11
- Ports .... 12
- Parallel Peripheral Interface (PPI) .... 12
- Dynamic Power Management .... 13
- Voltage Regulation .... 14
- Clock Signals .... 14
- Booting Modes .... 16
- Instruction Set Description .... 16
- Development Tools .... 17
- Designing an Emulator-Compatible Processor Board .. 18
- Related Documents .... 18

Pin Descriptions .... 19

Specifications .... 24
- Operating Conditions .... 24
- Electrical Characteristics .... 25
- Absolute Maximum Ratings .... 26
- ESD Sensitivity .... 26
- Package Information .... 26
- Timing Specifications .... 27
  - Asynchronous Memory Read Cycle Timing .... 29
  - Asynchronous Memory Write Cycle Timing .... 30
  - External Port Bus Request and Grant Cycle Timing .. 31
  - SDRAM Interface Timing .... 32
  - External DMA Request Timing .... 33
  - Parallel Peripheral Interface Timing .... 34
  - Serial Ports .... 37
  - Serial Peripheral Interface Port—Master Timing .... 41
  - Serial Peripheral Interface Port—Slave Timing .... 42
  - Universal Asynchronous Receiver-Transmitter (UART) Ports—Receive and Transmit Timing .... 43
  - General-Purpose Port Timing .... 44
  - Timer Cycle Timing .... 45
  - Timer Clock Timing .... 46
  - JTAG Test and Emulation Port Timing .... 47
  - 10/100 Ethernet MAC Controller Timing .... 48
- Output Drive Currents .... 51
- Power Dissipation .... 54
- Test Conditions .... 55
- Capacitive Loading .... 56
- Thermal Characteristics .... 59

182-Ball Mini-BGA Pinout .... 60

208-Ball Sparse Mini-BGA Pinout .... 63

Outline Dimensions .... 66
- Surface Mount Design .... 67

Ordering Guide .... 67


## REVISION HISTORY

**1/07—Revision C**

Revised Hibernate Operating Mode—Maximum Static Power Savings .... 13

Revised Figure 5, Voltage Regulator Circuit .... 14

Revised Pull-Up/Pull-Down on PH6 – GPIO/MII PHY-INT/RMII MDINT .... 21

Added 105°C data to Operating Conditions .... 24

Added 120°C $T_J$ data to Table 14, Core Clock Requirements—400 MHz Speed Grade .... 27

Revised Table 16, System Clock Requirements .... 28

Revised Table 17, Clock Input and Reset Timing .... 28

Added 105°C data and pin range for DATA and ADDR pins to table and figure in SDRAM Interface Timing .... 32

Removed ADSP-BF534YBCZ-4B from Ordering Guide ... 67

# GENERAL DESCRIPTION

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors are members of the Blackfin family of products, incorporating the Analog Devices/Intel Micro Signal Architecture (MSA). Blackfin processors combine a dual-MAC state-of-the-art signal processing engine, the advantages of a clean, orthogonal RISC-like microprocessor instruction set, and single-instruction, multiple-data (SIMD) multimedia capabilities into a single instruction-set architecture.

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors are completely code and pin compatible. They differ only with respect to their performance, on-chip memory, and presence of the Ethernet MAC module. Specific performance, memory, and feature configurations are shown in Table 1.

**Table 1. Processor Comparison**

| **Features** | | **ADSP-BF534** | **ADSP-BF536** | **ADSP-BF537** |
|---|---|---|---|---|
| Ethernet MAC | | — | 1 | 1 |
| CAN | | 1 | 1 | 1 |
| TWI | | 1 | 1 | 1 |
| SPORTs | | 2 | 2 | 2 |
| UARTs | | 2 | 2 | 2 |
| SPI | | 1 | 1 | 1 |
| GP Timers | | 8 | 8 | 8 |
| Watchdog Timers | | 1 | 1 | 1 |
| RTC | | 1 | 1 | 1 |
| Parallel Peripheral Interface | | 1 | 1 | 1 |
| GPIOs | | 48 | 48 | 48 |
| Memory Configuration | L1 Instruction SRAM/Cache | 16K bytes | 16K bytes | 16K bytes |
| | L1 Instruction SRAM | 48K bytes | 48K bytes | 48K bytes |
| | L1 Data SRAM/Cache | 32K bytes | 32K bytes | 32K bytes |
| | L1 Data SRAM | 32K bytes | — | 32K bytes |
| | L1 Scratchpad | 4K bytes | 4K bytes | 4K bytes |
| | L3 Boot ROM | 2K bytes | 2K bytes | 2K bytes |
| Maximum Speed Grade | | 500 MHz | 400 MHz | 600 MHz |
| Package Options:<br>Sparse Mini-BGA<br>Mini-BGA | | <br>208-Ball<br>182-Ball | <br>208-Ball<br>182-Ball | <br>208-Ball<br>182-Ball |

By integrating a rich set of industry-leading system peripherals and memory, the Blackfin processors are the platform of choice for next-generation applications that require RISC-like programmability, multimedia support, and leading-edge signal processing in one integrated package.

## PORTABLE LOW POWER ARCHITECTURE

Blackfin processors provide world-class power management and performance. They are produced with a low power and low voltage design methodology and feature on-chip dynamic power management, which is the ability to vary both the voltage and frequency of operation to significantly lower overall power consumption. This capability can result in a substantial reduction in power consumption, compared with just varying the frequency of operation. This allows longer battery life for portable appliances.

## SYSTEM INTEGRATION

The Blackfin processor is a highly integrated system-on-a-chip solution for the next generation of embedded network-connected applications. By combining industry-standard interfaces with a high performance signal processing core, cost-effective applications can be developed quickly, without the need for costly external components. The system peripherals include an IEEE-compliant 802.3 10/100 Ethernet MAC (ADSP-BF536 and ADSP-BF537 only), a CAN 2.0B controller, a TWI controller, two UART ports, an SPI port, two serial ports (SPORTs), nine general-purpose 32-bit timers (eight with PWM capability), a real-time clock, a watchdog timer, and a parallel peripheral interface (PPI).

## BLACKFIN PROCESSOR PERIPHERALS

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors contains a rich set of peripherals connected to the core via several high bandwidth buses, providing flexibility in system configuration as well as excellent overall system performance (see the block diagram on Page 1). The processors contain dedicated network communication modules and high speed serial and parallel ports, an interrupt controller for flexible management of interrupts from the on-chip peripherals or external sources, and power management control functions to tailor the performance and power characteristics of the processor and system to many application scenarios.

All of the peripherals, except for the general-purpose I/O, CAN, TWI, real-time clock, and timers, are supported by a flexible DMA structure. There are also separate memory DMA channels dedicated to data transfers between the processor's various memory spaces, including external SDRAM and asynchronous memory. Multiple on-chip buses running at up to 133 MHz provide enough bandwidth to keep the processor core running along with activity on all of the on-chip and external peripherals.

The Blackfin processors include an on-chip voltage regulator in support of the processors' dynamic power management capability. The voltage regulator provides a range of core voltage levels when supplied from a single 2.25 V to 3.6 V input. The voltage regulator can be bypassed at the user's discretion.

## BLACKFIN PROCESSOR CORE

As shown in Figure 2 on Page 4, the Blackfin processor core contains two 16-bit multipliers, two 40-bit accumulators, two 40-bit ALUs, four video ALUs, and a 40-bit shifter. The computation units process 8-, 16-, or 32-bit data from the register file.

The compute register file contains eight 32-bit registers. When performing compute operations on 16-bit operand data, the register file operates as 16 independent 16-bit registers. All operands for compute operations come from the multiported register file and instruction constant fields.

Each MAC can perform a 16-bit by 16-bit multiply in each cycle, accumulating the results into the 40-bit accumulators. Signed and unsigned formats, rounding, and saturation are supported.

The ALUs perform a traditional set of arithmetic and logical operations on 16-bit or 32-bit data. In addition, many special instructions are included to accelerate various signal processing tasks. These include bit operations such as field extract and population count, modulo $2^{32}$ multiply, divide primitives, saturation and rounding, and sign/exponent detection. The set of video instructions include byte alignment and packing operations, 16-bit and 8-bit adds with clipping, 8-bit average operations, and 8-bit subtract/absolute value/accumulate (SAA) operations. Also provided are the compare/select and vector search instructions.

For certain instructions, two 16-bit ALU operations can be performed simultaneously on register pairs (a 16-bit high half and 16-bit low half of a compute register). If the second ALU is used, quad 16-bit operations are possible.

The 40-bit shifter can perform shifts and rotates and is used to support normalization, field extract, and field deposit instructions.

The program sequencer controls the flow of instruction execution, including instruction alignment and decoding. For program flow control, the sequencer supports PC relative and indirect conditional jumps (with static branch prediction), and subroutine calls. Hardware is provided to support zero-overhead looping. The architecture is fully interlocked, meaning that the programmer need not manage the pipeline when executing instructions with data dependencies.

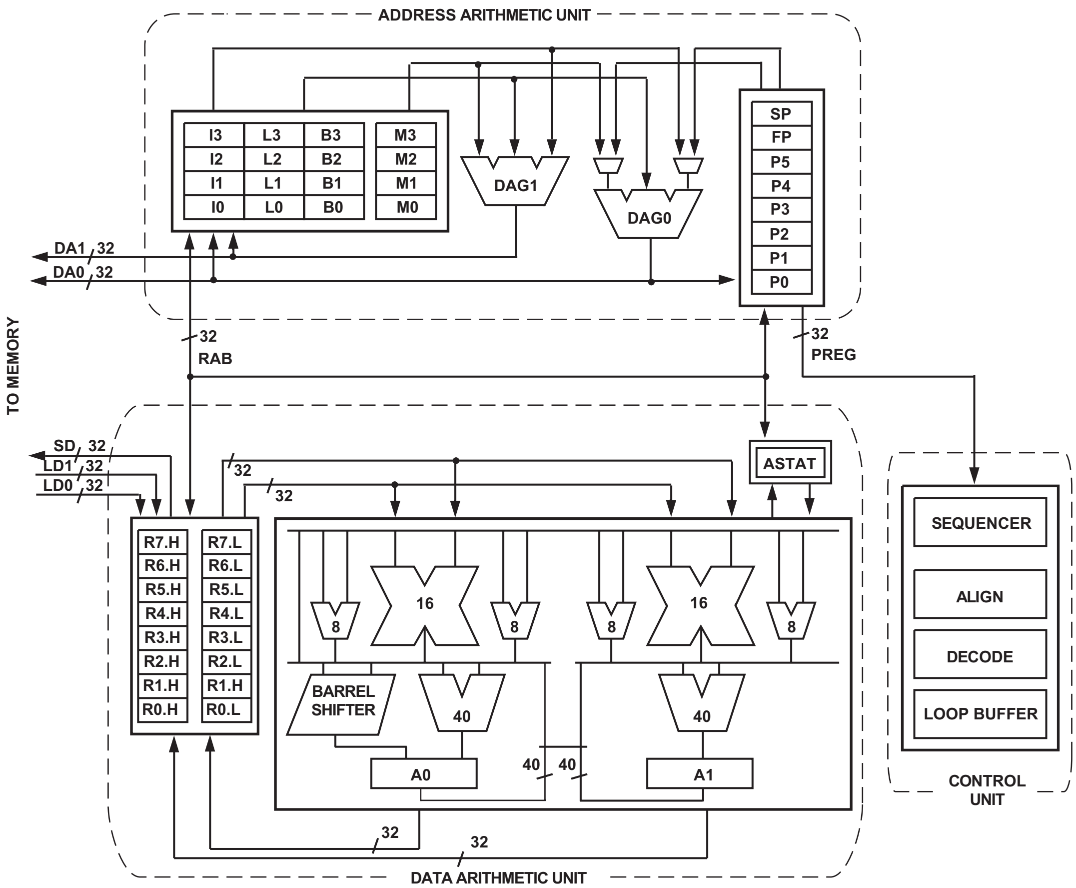



*Figure 2. Blackfin Processor Core*

The address arithmetic unit provides two addresses for simultaneous dual fetches from memory. It contains a multiported register file consisting of four sets of 32-bit index, modify, length, and base registers (for circular buffering), and eight additional 32-bit pointer registers (for C-style indexed stack manipulation).

Blackfin processors support a modified Harvard architecture in combination with a hierarchical memory structure. Level 1 (L1) memories are those that typically operate at the full processor speed with little or no latency. At the L1 level, the instruction memory holds instructions only. The two data memories hold data, and a dedicated scratchpad data memory stores stack and local variable information.

In addition, multiple L1 memory blocks are provided, offering a configurable mix of SRAM and cache. The memory management unit (MMU) provides memory protection for individual tasks that may be operating on the core and can protect system registers from unintended access.

The architecture provides three modes of operation: user mode, supervisor mode, and emulation mode. User mode has restricted access to certain system resources, thus providing a protected software environment, while supervisor mode has unrestricted access to the system and core resources.

The Blackfin processor instruction set has been optimized so that 16-bit opcodes represent the most frequently used instructions, resulting in excellent compiled code density. Complex DSP instructions are encoded into 32-bit opcodes, representing fully featured multifunction instructions. Blackfin processors support a limited multi-issue capability, where a 32-bit instruction can be issued in parallel with two 16-bit instructions, allowing the programmer to use many of the core resources in a single instruction cycle.

The Blackfin processor assembly language uses an algebraic syntax for ease of coding and readability. The architecture has been optimized for use in conjunction with the C/C++ compiler, resulting in fast and efficient software implementations.

## MEMORY ARCHITECTURE

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors view memory as a single unified 4G byte address space, using 32-bit addresses. All resources, including internal memory, external memory, and I/O control registers, occupy separate sections of this common address space. The memory portions of this address space are arranged in a hierarchical structure to provide a good cost/performance balance of some very fast, low latency on-chip memory as cache or SRAM, and larger, lower cost, and performance off-chip memory systems. See Figure 3.

The on-chip L1 memory system is the highest performance memory available to the Blackfin processor. The off-chip memory system, accessed through the external bus interface unit (EBIU), provides expansion with SDRAM, flash memory, and SRAM, optionally accessing up to 516M bytes of physical memory.

The memory DMA controller provides high bandwidth data-movement capability. It can perform block transfers of code or data between the internal memory and the external memory spaces.

### *Internal (On-Chip) Memory*

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors have three blocks of on-chip memory providing high-bandwidth access to the core.

The first block is the L1 instruction memory, consisting of 64K bytes SRAM, of which 16K bytes can be configured as a four-way set-associative cache. This memory is accessed at full processor speed.

The second on-chip memory block is the L1 data memory, consisting of up to two banks of up to 32K bytes each. Each memory bank is configurable, offering both cache and SRAM functionality. This memory block is accessed at full processor speed.

The third memory block is a 4K byte scratchpad SRAM, which runs at the same speed as the L1 memories, but is only accessible as data SRAM, and cannot be configured as cache memory.

### *External (Off-Chip) Memory*

External memory is accessed via the EBIU. This 16-bit interface provides a glueless connection to a bank of synchronous DRAM (SDRAM) as well as up to four banks of asynchronous memory devices including flash, EPROM, ROM, SRAM, and memory mapped I/O devices.

The PC133-compliant SDRAM controller can be programmed to interface to up to 512M bytes of SDRAM. A separate row can be open for each SDRAM internal bank, and the SDRAM controller supports up to 4 internal SDRAM banks, improving overall performance.

The asynchronous memory controller can be programmed to control up to four banks of devices with very flexible timing parameters for a wide variety of devices. Each bank occupies a 1M byte segment regardless of the size of the devices used, so that these banks are only contiguous if each is fully populated with 1M byte of memory.

### *I/O Memory Space*

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors do not define a separate I/O space. All resources are mapped through the flat 32-bit address space. On-chip I/O devices have their control registers mapped into memory-mapped registers (MMRs) at addresses near the top of the 4G byte address space. These are separated into two smaller blocks, one which contains the control MMRs for all core functions, and the other which contains the registers needed for setup and control of the on-chip peripherals outside of the core. The MMRs are accessible only in supervisor mode and appear as reserved space to on-chip peripherals.

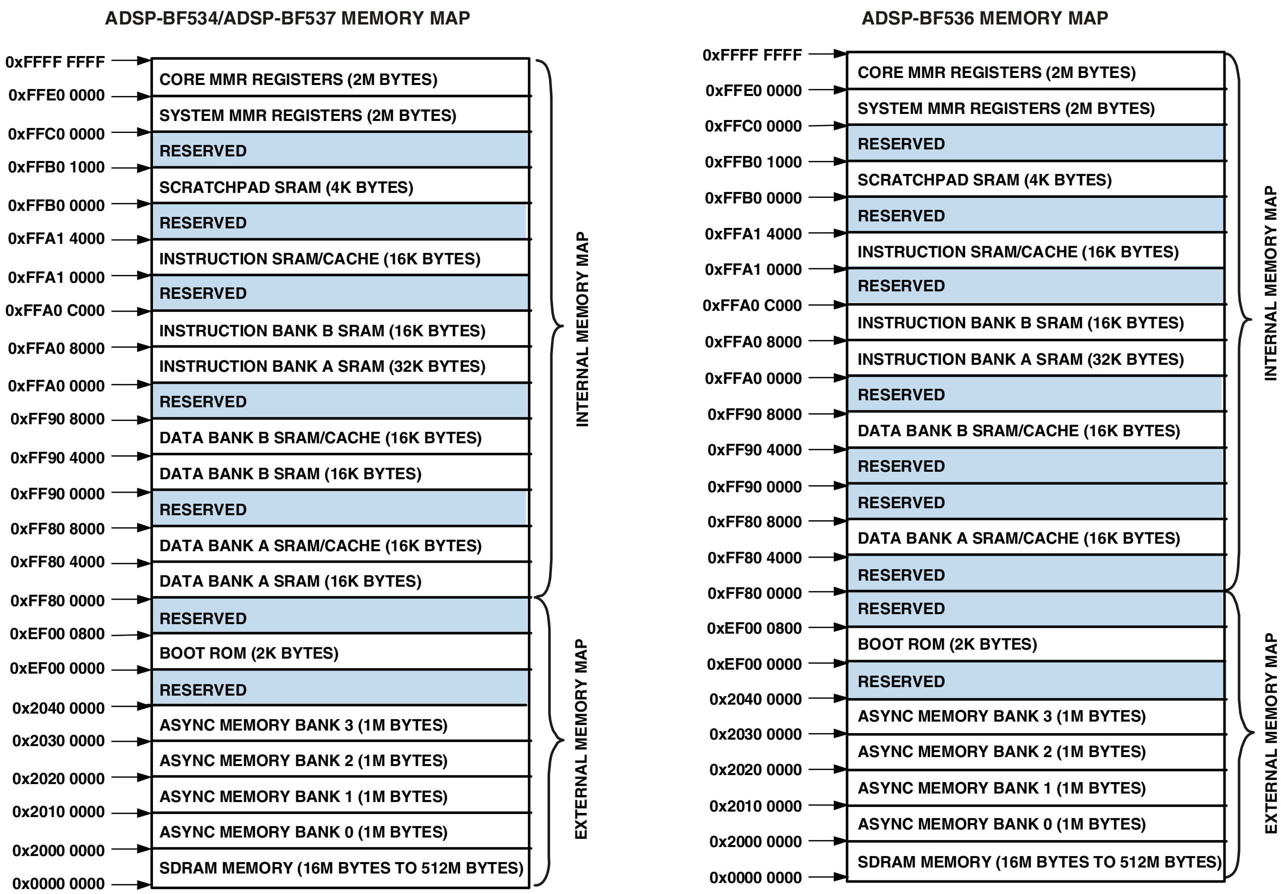



*Figure 3. ADSP-BF534/ADSP-BF536/ADSP-BF537 Memory Maps*

### *Booting*

The Blackfin processor contains a small on-chip boot kernel, which configures the appropriate peripheral for booting. If the Blackfin processor is configured to boot from boot ROM memory space, the processor starts executing from the on-chip boot ROM. For more information, see Booting Modes on Page 16.

### *Event Handling*

The event controller on the Blackfin processor handles all asynchronous and synchronous events to the processor. The Blackfin processor provides event handling that supports both nesting and prioritization. Nesting allows multiple event service routines to be active simultaneously. Prioritization ensures that servicing of a higher priority event takes precedence over servicing of a lower priority event. The controller provides support for five different types of events:

- Emulation – An emulation event causes the processor to enter emulation mode, allowing command and control of the processor via the JTAG interface.
- Reset – This event resets the processor.
- Nonmaskable Interrupt (NMI) – The NMI event can be generated by the software watchdog timer or by the NMI input signal to the processor. The NMI event is frequently used as a power-down indicator to initiate an orderly shutdown of the system.
- Exceptions – Events that occur synchronously to program flow (in other words, the exception is taken before the instruction is allowed to complete). Conditions such as data alignment violations and undefined instructions cause exceptions.
- Interrupts – Events that occur asynchronously to program flow. They are caused by input pins, timers, and other peripherals, as well as by an explicit software instruction.

Each event type has an associated register to hold the return address and an associated return-from-event instruction. When an event is triggered, the state of the processor is saved on the supervisor stack.

The Blackfin processor event controller consists of two stages, the core event controller (CEC) and the system interrupt controller (SIC). The core event controller works with the system interrupt controller to prioritize and control all system events.

Conceptually, interrupts from the peripherals enter into the SIC, and are then routed directly into the general-purpose interrupts of the CEC.

### *Core Event Controller (CEC)*

The CEC supports nine general-purpose interrupts (IVG15–7), in addition to the dedicated interrupt and exception events. Of these general-purpose interrupts, the two lowest priority interrupts (IVG15–14) are recommended to be reserved for software interrupt handlers, leaving seven prioritized interrupt inputs to support the peripherals of the Blackfin processor. Table 2 describes the inputs to the CEC, identifies their names in the event vector table (EVT), and lists their priorities.

**Table 2. Core Event Controller (CEC)**

| Priority (0 Is Highest) | Event Class | EVT Entry |
|---|---|---|
| 0 | Emulation/Test Control | EMU |
| 1 | Reset | RST |
| 2 | Nonmaskable Interrupt | NMI |
| 3 | Exception | EVX |
| 4 | Reserved | — |
| 5 | Hardware Error | IVHW |
| 6 | Core Timer | IVTMR |
| 7 | General-Purpose Interrupt 7 | IVG7 |
| 8 | General-Purpose Interrupt 8 | IVG8 |
| 9 | General-Purpose Interrupt 9 | IVG9 |
| 10 | General-Purpose Interrupt 10 | IVG10 |
| 11 | General-Purpose Interrupt 11 | IVG11 |
| 12 | General-Purpose Interrupt 12 | IVG12 |
| 13 | General-Purpose Interrupt 13 | IVG13 |
| 14 | General-Purpose Interrupt 14 | IVG14 |
| 15 | General-Purpose Interrupt 15 | IVG15 |

### *System Interrupt Controller (SIC)*

The system interrupt controller provides the mapping and routing of events from the many peripheral interrupt sources to the prioritized general-purpose interrupt inputs of the CEC. Although the processor provides a default mapping, the user can alter the mappings and priorities of interrupt events by writing the appropriate values into the interrupt assignment registers (IAR). Table 3 describes the inputs into the SIC and the default mappings into the CEC.

**Table 3. System Interrupt Controller (SIC)**

| Peripheral Interrupt Event | Default Mapping | Peripheral Interrupt ID |
|---|---|---|
| PLL Wakeup | IVG7 | 0 |
| DMA Error (generic) | IVG7 | 1 |
| DMAR0 Block Interrupt | IVG7 | 1 |
| DMAR1 Block Interrupt | IVG7 | 1 |
| DMAR0 Overflow Error | IVG7 | 1 |
| DMAR1 Overflow Error | IVG7 | 1 |
| CAN Error | IVG7 | 2 |
| Ethernet Error (ADSP-BF536 and ADSP-BF537 only) | IVG7 | 2 |
| SPORT 0 Error | IVG7 | 2 |
| SPORT 1 Error | IVG7 | 2 |
| PPI Error | IVG7 | 2 |
| SPI Error | IVG7 | 2 |
| UART0 Error | IVG7 | 2 |
| UART1 Error | IVG7 | 2 |
| Real-Time Clock | IVG8 | 3 |
| DMA Channel 0 (PPI) | IVG8 | 4 |
| DMA Channel 3 (SPORT 0 Rx) | IVG9 | 5 |
| DMA Channel 4 (SPORT 0 Tx) | IVG9 | 6 |
| DMA Channel 5 (SPORT 1 Rx) | IVG9 | 7 |
| DMA Channel 6 (SPORT 1 Tx) | IVG9 | 8 |
| TWI | IVG10 | 9 |
| DMA Channel 7 (SPI) | IVG10 | 10 |
| DMA Channel 8 (UART0 Rx) | IVG10 | 11 |
| DMA Channel 9 (UART0 Tx) | IVG10 | 12 |
| DMA Channel 10 (UART1 Rx) | IVG10 | 13 |
| DMA Channel 11 (UART1 Tx) | IVG10 | 14 |
| CAN Rx | IVG11 | 15 |
| CAN Tx | IVG11 | 16 |
| DMA Channel 1 (Ethernet Rx, ADSP-BF536 and ADSP-BF537 only) | IVG11 | 17 |
| Port H Interrupt A | IVG11 | 17 |
| DMA Channel 2 (Ethernet Tx, ADSP-BF536 and ADSP-BF537 only) | IVG11 | 18 |
| Port H Interrupt B | IVG11 | 18 |
| Timer 0 | IVG12 | 19 |
| Timer 1 | IVG12 | 20 |
| Timer 2 | IVG12 | 21 |
| Timer 3 | IVG12 | 22 |
| Timer 4 | IVG12 | 23 |
| Timer 5 | IVG12 | 24 |
| Timer 6 | IVG12 | 25 |
| Timer 7 | IVG12 | 26 |
| Port F, G Interrupt A | IVG12 | 27 |
| Port G Interrupt B | IVG12 | 28 |

**Table 3. System Interrupt Controller (SIC) (Continued)**

| Peripheral Interrupt Event | Default Mapping | Peripheral Interrupt ID |
|---|---|---|
| DMA Channels 12 and 13 (Memory DMA Stream 0) | IVG13 | 29 |
| DMA Channels 14 and 15 (Memory DMA Stream 1) | IVG13 | 30 |
| Software Watchdog Timer | IVG13 | 31 |
| Port F Interrupt B | IVG13 | 31 |

### *Event Control*

The Blackfin processor provides a very flexible mechanism to control the processing of events. In the CEC, three registers are used to coordinate and control events. Each register is 16 bits wide:

- CEC interrupt latch register (ILAT) – Indicates when events have been latched. The appropriate bit is set when the processor has latched the event and cleared when the event has been accepted into the system. This register is updated automatically by the controller, but it may be written only when its corresponding IMASK bit is cleared.
- CEC interrupt mask register (IMASK) – Controls the masking and unmasking of individual events. When a bit is set in the IMASK register, that event is unmasked and is processed by the CEC when asserted. A cleared bit in the IMASK register masks the event, preventing the processor from servicing the event even though the event may be latched in the ILAT register. This register may be read or written while in supervisor mode. (Note that general-purpose interrupts can be globally enabled and disabled with the STI and CLI instructions, respectively.)
- CEC interrupt pending register (IPEND) – The IPEND register keeps track of all nested events. A set bit in the IPEND register indicates the event is currently active or nested at some level. This register is updated automatically by the controller but may be read while in supervisor mode.

The SIC allows further control of event processing by providing three 32-bit interrupt control and status registers. Each register contains a bit corresponding to each of the peripheral interrupt events shown in Table 3 on Page 7.

- SIC interrupt mask register (SIC_IMASK) – Controls the masking and unmasking of each peripheral interrupt event. When a bit is set in the register, that peripheral event is unmasked and is processed by the system when asserted. A cleared bit in the register masks the peripheral event, preventing the processor from servicing the event.
- SIC interrupt status register (SIC_ISR) – As multiple peripherals can be mapped to a single event, this register allows the software to determine which peripheral event source triggered the interrupt. A set bit indicates the peripheral is asserting the interrupt, and a cleared bit indicates the peripheral is not asserting the event.
- SIC interrupt wakeup enable register (SIC_IWR) – By enabling the corresponding bit in this register, a peripheral can be configured to wake up the processor, should the core be idled when the event is generated. (For more information, see Dynamic Power Management on Page 13.)

Because multiple interrupt sources can map to a single general-purpose interrupt, multiple pulse assertions can occur simultaneously, before or during interrupt processing for an interrupt event already detected on this interrupt input. The IPEND register contents are monitored by the SIC as the interrupt acknowledgement.

The appropriate ILAT register bit is set when an interrupt rising edge is detected (detection requires two core clock cycles). The bit is cleared when the respective IPEND register bit is set. The IPEND bit indicates that the event has entered into the processor pipeline. At this point the CEC recognizes and queues the next rising edge event on the corresponding event input. The minimum latency from the rising edge transition of the general-purpose interrupt to the IPEND output asserted is three core clock cycles; however, the latency can be much higher, depending on the activity within and the state of the processor.

## DMA CONTROLLERS

The Blackfin processors have multiple, independent DMA controllers that support automated data transfers with minimal overhead for the processor core. DMA transfers can occur between the processor's internal memories and any of its DMA-capable peripherals. Additionally, DMA transfers can be accomplished between any of the DMA-capable peripherals and external devices connected to the external memory interfaces, including the SDRAM controller and the asynchronous memory controller. DMA-capable peripherals include the Ethernet MAC (ADSP-BF536 and ADSP-BF537 only), SPORTs, SPI port, UARTs, and PPI. Each individual DMA-capable peripheral has at least one dedicated DMA channel.

The DMA controller supports both one-dimensional (1-D) and two-dimensional (2-D) DMA transfers. DMA transfer initialization can be implemented from registers or from sets of parameters called descriptor blocks.

The 2-D DMA capability supports arbitrary row and column sizes up to 64K elements by 64K elements, and arbitrary row and column step sizes up to ±32K elements. Furthermore, the column step size can be less than the row step size, allowing implementation of interleaved data streams. This feature is especially useful in video applications where data can be de-interleaved on the fly.

Examples of DMA types supported by the DMA controller include:

- A single, linear buffer that stops upon completion
- A circular, auto-refreshing buffer that interrupts on each full or fractionally full buffer
- 1-D or 2-D DMA using a linked list of descriptors
- 2-D DMA using an array of descriptors, specifying only the base DMA address within a common page.

In addition to the dedicated peripheral DMA channels, there are two memory DMA channels provided for transfers between the various memories of the processor system. This enables transfers of blocks of data between any of the memories—including external SDRAM, ROM, SRAM, and flash memory—with minimal processor intervention. Memory DMA transfers can be controlled by a very flexible descriptor-based methodology or by a standard register-based autobuffer mechanism.

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors also have an external DMA controller capability via dual external DMA request pins when used in conjunction with the external bus interface unit (EBIU). This functionality can be used when a high speed interface is required for external FIFOs and high bandwidth communications peripherals such as USB 2.0. It allows control of the number of data transfers for memDMA. The number of transfers per edge is programmable. This feature can be programmed to allow memDMA to have an increased priority on the external bus relative to the core.

## REAL-TIME CLOCK

The real-time clock (RTC) provides a robust set of digital watch features, including current time, stopwatch, and alarm. The RTC is clocked by a 32.768 kHz crystal external to the processor. The RTC peripheral has dedicated power supply pins so that it can remain powered up and clocked even when the rest of the processor is in a low-power state. The RTC provides several programmable interrupt options, including interrupt per second, minute, hour, or day clock ticks, interrupt on programmable stopwatch countdown, or interrupt at a programmed alarm time.

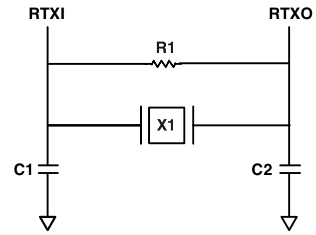
The 32.768 kHz input clock frequency is divided down to a 1 Hz signal by a prescaler. The counter function of the timer consists of four counters: a 60-second counter, a 60-minute counter, a 24-hour counter, and an 32,768-day counter.

When enabled, the alarm function generates an interrupt when the output of the timer matches the programmed value in the alarm control register. There are two alarms: The first alarm is for a time of day, while the second alarm is for a day and time of that day.

The stopwatch function counts down from a programmed value, with one-second resolution. When the stopwatch is enabled and the counter underflows, an interrupt is generated.

Like the other peripherals, the RTC can wake up the processor from sleep mode upon generation of any RTC wakeup event. Additionally, an RTC wakeup event can wake up the processor from deep sleep mode, and wake up the on-chip internal voltage regulator from the hibernate operating mode.

Connect RTC pins RTXI and RTXO with external components as shown in Figure 4.




SUGGESTED COMPONENTS:
ECLIPTEK EC38J (THROUGH-HOLE PACKAGE)
EPSON MC405 12pF LOAD (SURFACE MOUNT PACKAGE)
C1 = 22pF
C2 = 22pF
R1 = 10MΩ

NOTE: C1 AND C2 ARE SPECIFIC TO CRYSTAL SPECIFIED FOR X1. CONTACT CRYSTAL MANUFACTURER FOR DETAILS. C1 AND C2 SPECIFICATIONS ASSUME BOARD TRACE CAPACITANCE OF 3pF.

*Figure 4. External Components for RTC*

## WATCHDOG TIMER

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors include a 32-bit timer that can be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state through generation of a hardware reset, nonmaskable interrupt (NMI), or general-purpose interrupt, if the timer expires before being reset by software. The programmer initializes the count value of the timer, enables the appropriate interrupt, then enables the timer. Thereafter, the software must reload the counter before it counts to zero from the programmed value. This protects the system from remaining in an unknown state where software, which would normally reset the timer, has stopped running due to an external noise condition or software error.

If configured to generate a hardware reset, the watchdog timer resets both the core and the processor peripherals. After a reset, software can determine if the watchdog was the source of the hardware reset by interrogating a status bit in the watchdog timer control register.

The timer is clocked by the system clock (SCLK), at a maximum frequency of $f_{SCLK}$.

## TIMERS

There are nine general-purpose programmable timer units in the processor. Eight timers have an external pin that can be configured either as a pulse width modulator (PWM) or timer output, as an input to clock the timer, or as a mechanism for measuring pulse widths and periods of external events. These timers can be synchronized to an external clock input to the several other associated PF pins, to an external clock input to the PPI_CLK input pin, or to the internal SCLK.

The timer units can be used in conjunction with the two UARTs and the CAN controller to measure the width of the pulses in the data stream to provide a software auto-baud detect function for the respective serial channels.

The timers can generate interrupts to the processor core providing periodic events for synchronization, either to the system clock or to a count of external signals.

In addition to the eight general-purpose programmable timers, a ninth timer is also provided. This extra timer is clocked by the internal processor clock and is typically used as a system tick clock for generating periodic interrupts in an operating system.

## SERIAL PORTS (SPORTs)

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors incorporate two dual-channel synchronous serial ports (SPORT0 and SPORT1) for serial and multiprocessor communications. The SPORTs support the following features:

- $I^2S$ capable operation.
- Bidirectional operation – Each SPORT has two sets of independent transmit and receive pins, enabling eight channels of $I^2S$ stereo audio.
- Buffered (8-deep) transmit and receive ports – Each port has a data register for transferring data words to and from other processor components and shift registers for shifting data in and out of the data registers.
- Clocking – Each transmit and receive port can either use an external serial clock or generate its own, in frequencies ranging from ($f_{SCLK}$/131,070) Hz to ($f_{SCLK}$/2) Hz.
- Word length – Each SPORT supports serial data words from 3 to 32 bits in length, transferred most significant bit first or least significant bit first.
- Framing – Each transmit and receive port can run with or without frame sync signals for each data word. Frame sync signals can be generated internally or externally, active high or low, and with either of two pulse widths and early or late frame sync.
- Companding in hardware – Each SPORT can perform A-law or μ-law companding according to ITU recommendation G.711. Companding can be selected on the transmit and/or receive channel of the SPORT without additional latencies.
- DMA operations with single-cycle overhead – Each SPORT can automatically receive and transmit multiple buffers of memory data. The processor can link or chain sequences of DMA transfers between a SPORT and memory.
- Interrupts – Each transmit and receive port generates an interrupt upon completing the transfer of a data word or after transferring an entire data buffer, or buffers, through DMA.
- Multichannel capability – Each SPORT supports 128 channels out of a 1024-channel window and is compatible with the H.100, H.110, MVIP-90, and HMVIP standards.

## SERIAL PERIPHERAL INTERFACE (SPI) PORT

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors have an SPI-compatible port that enables the processor to communicate with multiple SPI-compatible devices.

The SPI interface uses three pins for transferring data: two data pins (Master Output-Slave Input, MOSI, and Master Input-Slave Output, MISO) and a clock pin (serial clock, SCK). An SPI chip select input pin ($\overline{\text{SPISS}}$) lets other SPI devices select the processor, and seven SPI chip select output pins ($\overline{\text{SPISEL7–1}}$) let the processor select other SPI devices. The SPI select pins are reconfigured programmable flag pins. Using these pins, the SPI port provides a full-duplex, synchronous serial interface, which supports both master/slave modes and multimaster environments.

The SPI port's baud rate and clock phase/polarities are programmable, and it has an integrated DMA controller, configurable to support transmit or receive data streams. The SPI's DMA controller can only service unidirectional accesses at any given time.

The SPI port's clock rate is calculated as:

$$SPI\ Clock\ Rate = \frac{f_{SCLK}}{2 \times SPI_Baud}$$

Where the 16-bit SPI_Baud register contains a value of 2 to 65,535.

During transfers, the SPI port simultaneously transmits and receives by serially shifting data in and out on its two serial data lines. The serial clock line synchronizes the shifting and sampling of data on the two serial data lines.

## UART PORTS

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors provide two full-duplex universal asynchronous receiver and transmitter (UART) ports, which are fully compatible with PC-standard UARTs. Each UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. A UART port includes support for five to eight data bits, one or two stop bits, and none, even, or odd parity. Each UART port supports two modes of operation:

- PIO (programmed I/O) – The processor sends or receives data by writing or reading I/O mapped UART registers. The data is double-buffered on both transmit and receive.
- DMA (direct memory access) – The DMA controller transfers both transmit and receive data. This reduces the number and frequency of interrupts required to transfer data to and from memory. The UART has two dedicated DMA channels, one for transmit and one for receive. These DMA channels have lower default priority than most DMA channels because of their relatively low service rates.

Each UART port's baud rate, serial data format, error code generation and status, and interrupts are programmable:

- Supporting bit rates ranging from ($f_{SCLK}$/1,048,576) to ($f_{SCLK}$/16) bits per second.
- Supporting data formats from 7 to 12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

The UART port's clock rate is calculated as:

$$UART\ Clock\ Rate = \frac{f_{SCLK}}{16 \times UART_Divisor}$$

Where the 16-bit UARTx_Divisor comes from the DLH register (most significant 8 bits) and UARTx_DLL register (least significant 8 bits).

In conjunction with the general-purpose timer functions, autobaud detection is supported.

The capabilities of the UARTs are further extended with support for the infrared data association (IrDA) serial infrared physical layer link specification (SIR) protocol.

## CONTROLLER AREA NETWORK (CAN)

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors offer a CAN controller that is a communication controller implementing the CAN 2.0B (active) protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. The CAN protocol is well-suited for control applications due to its capability to communicate reliably over a network, since the protocol incorporates CRC checking message error tracking, and fault node confinement.

The CAN controller offers the following features:

- 32 mailboxes (eight receive only, eight transmit only, 16 configurable for receive or transmit).
- Dedicated acceptance masks for each mailbox.
- Additional data filtering on first two bytes.
- Support for both the standard (11-bit) and extended (29-bit) identifier (ID) message formats.
- Support for remote frames.
- Active or passive network support.
- CAN wakeup from hibernation mode (lowest static power consumption mode).
- Interrupts, including: Tx complete, Rx complete, error, global.

The electrical characteristics of each network connection are very demanding so the CAN interface is typically divided into two parts: a controller and a transceiver. This allows a single controller to support different drivers and CAN networks. The CAN module represents only the controller part of the interface. The controller interface supports connection to 3.3 V high-speed, fault-tolerant, single-wire transceivers.

## TWI CONTROLLER INTERFACE

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors include a 2-wire interface (TWI) module for providing a simple exchange method of control data between multiple devices. The TWI is compatible with the widely used $I^2C$® bus standard. The TWI module offers the capabilities of simultaneous master and slave operation, support for both 7-bit addressing and multimedia data arbitration. The TWI interface utilizes two pins for transferring clock (SCL) and data (SDA) and supports the protocol at speeds up to 400 kbps. The TWI interface pins are compatible with 5 V logic levels.

Additionally, the processor's TWI module is fully compatible with serial camera control bus (SCCB) functionality for easier control of various CMOS camera sensor devices.

## 10/100 ETHERNET MAC

The ADSP-BF536 and ADSP-BF537 processors offer the capability to directly connect to a network by way of an embedded fast Ethernet Media Access Controller (MAC) that supports both 10-BaseT (10 Mbps) and 100-BaseT (100 Mbps) operation. The 10/100 Ethernet MAC peripheral is fully compliant to the IEEE 802.3-2002 standard, and it provides programmable features designed to minimize supervision, bus use, or message processing by the rest of the processor system.

Some standard features are:

- Support of MII and RMII protocols for external PHYs.
- Full duplex and half duplex modes.
- Data framing and encapsulation: generation and detection of preamble, length padding, and FCS.
- Media access management (in half-duplex operation): collision and contention handling, including control of retransmission of collision frames and of back-off timing.
- Flow control (in full-duplex operation): generation and detection of PAUSE frames.
- Station management: generation of MDC/MDIO frames for read-write access to PHY registers.
- SCLK operating range down to 25 MHz (active and sleep operating modes).
- Internal loopback from Tx to Rx.

Some advanced features are:

- Buffered crystal output to external PHY for support of a single crystal system.
- Automatic checksum computation of IP header and IP payload fields of Rx frames.
- Independent 32-bit descriptor-driven Rx and Tx DMA channels.
- Frame status delivery to memory via DMA, including frame completion semaphores, for efficient buffer queue management in software.
- Tx DMA support for separate descriptors for MAC header and payload to eliminate buffer copy operations.
- Convenient frame alignment modes support even 32-bit alignment of encapsulated Rx or Tx IP packet data in memory after the 14-byte MAC header.
- Programmable Ethernet event interrupt supports any combination of:
  - Any selected Rx or Tx frame status conditions.
  - PHY interrupt condition.
  - Wakeup frame detected.
  - Any selected MAC management counter(s) at half-full.
  - DMA descriptor error.
- 47 MAC management statistics counters with selectable clear-on-read behavior and programmable interrupts on half maximum value.

- Programmable Rx address filters, including a 64-bit address hash table for multicast and/or unicast frames, and programmable filter modes for broadcast, multicast, unicast, control, and damaged frames.
- Advanced power management supporting unattended transfer of Rx and Tx frames and status to/from external memory via DMA during low-power sleep mode.
- System wakeup from sleep operating mode upon magic packet or any of four user-definable wakeup frame filters.
- Support for 802.3Q tagged VLAN frames.
- Programmable MDC clock rate and preamble suppression.
- In RMII operation, 7 unused pins may be configured as GPIO pins for other purposes.

## PORTS

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors group the many peripheral signals to four ports—Port F, Port G, Port H, and Port J. Most of the associated pins are shared by multiple signals. The ports function as multiplexer controls. Eight of the pins (Port F7–0) offer high source/high sink current capabilities.

### *General-Purpose I/O (GPIO)*

The processors have 48 bidirectional, general-purpose I/O (GPIO) pins allocated across three separate GPIO modules—PORTFIO, PORTGIO, and PORTHIO, associated with Port F, Port G, and Port H, respectively. Port J does not provide GPIO functionality. Each GPIO-capable pin shares functionality with other processor peripherals via a multiplexing scheme; however, the GPIO functionality is the default state of the device upon power-up. Neither GPIO output or input drivers are active by default. Each general-purpose port pin can be individually controlled by manipulation of the port control, status, and interrupt registers:

- GPIO direction control register – Specifies the direction of each individual GPIO pin as input or output.
- GPIO control and status registers – The processors employ a "write one to modify" mechanism that allows any combination of individual GPIO pins to be modified in a single instruction, without affecting the level of any other GPIO pins. Four control registers are provided. One register is written in order to set pin values, one register is written in order to clear pin values, one register is written in order to toggle pin values, and one register is written in order to specify a pin value. Reading the GPIO status register allows software to interrogate the sense of the pins.
- GPIO interrupt mask registers – The two GPIO interrupt mask registers allow each individual GPIO pin to function as an interrupt to the processor. Similar to the two GPIO control registers that are used to set and clear individual pin values, one GPIO interrupt mask register sets bits to enable interrupt function, and the other GPIO interrupt mask register clears bits to disable interrupt function. GPIO pins defined as inputs can be configured to generate hardware interrupts, while output pins can be triggered by software interrupts.
- GPIO interrupt sensitivity registers – The two GPIO interrupt sensitivity registers specify whether individual pins are level- or edge-sensitive and specify—if edge-sensitive—whether just the rising edge or both the rising and falling edges of the signal are significant. One register selects the type of sensitivity, and one register selects which edges are significant for edge-sensitivity.

## PARALLEL PERIPHERAL INTERFACE (PPI)

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors provide a parallel peripheral interface (PPI) that can connect directly to parallel A/D and D/A converters, ITU-R-601/656 video encoders and decoders, and other general-purpose peripherals. The PPI consists of a dedicated input clock pin, up to 3 frame synchronization pins, and up to 16 data pins.

In ITU-R-656 modes, the PPI receives and parses a data stream of 8-bit or 10-bit data elements. On-chip decode of embedded preamble control and synchronization information is supported.

Three distinct ITU-R-656 modes are supported:

- Active video only mode – The PPI does not read in any data between the End of Active Video (EAV) and Start of Active Video (SAV) preamble symbols, or any data present during the vertical blanking intervals. In this mode, the control byte sequences are not stored to memory; they are filtered by the PPI.
- Vertical blanking only mode – The PPI only transfers vertical blanking interval (VBI) data, as well as horizontal blanking information and control byte sequences on VBI lines.
- Entire field mode – The entire incoming bitstream is read in through the PPI. This includes active video, control preamble sequences, and ancillary data that may be embedded in horizontal and vertical blanking intervals.

Though not explicitly supported, ITU-R-656 output functionality can be achieved by setting up the entire frame structure (including active video, blanking, and control information) in memory and streaming the data out the PPI in a frame sync-less mode. The processor's 2-D DMA features facilitate this transfer by allowing the static frame buffer (blanking and control codes) to be placed in memory once, and simply updating the active video information on a per-frame basis.

The general-purpose modes of the PPI are intended to suit a wide variety of data capture and transmission applications. The modes are divided into four main categories, each allowing up to 16 bits of data transfer per PPI_CLK cycle:

- Data receive with internally generated frame syncs
- Data receive with externally generated frame syncs
- Data transmit with internally generated frame syncs
- Data transmit with externally generated frame syncs

These modes support ADC/DAC connections, as well as video communication with hardware signalling. Many of the modes support more than one level of frame synchronization. If desired, a programmable delay can be inserted between assertion of a frame sync and reception/transmission of data.

## DYNAMIC POWER MANAGEMENT

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processors provide five operating modes, each with a different performance and power profile. In addition, dynamic power management provides the control functions to dynamically alter the processor core supply voltage, further reducing power dissipation. Control of clocking to each of the peripherals also reduces power consumption. See Table 4 for a summary of the power settings for each mode.

### *Full-On Operating Mode—Maximum Performance*

In the full-on mode, the PLL is enabled and is not bypassed, providing capability for maximum operational frequency. This is the power-up default execution state in which maximum performance can be achieved. The processor core and all enabled peripherals run at full speed.

### *Active Operating Mode—Moderate Power Savings*

In the active mode, the PLL is enabled but bypassed. Because the PLL is bypassed, the processor's core clock (CCLK) and system clock (SCLK) run at the input clock (CLKIN) frequency. In this mode, the CLKIN to CCLK multiplier ratio can be changed, although the changes are not realized until the full-on mode is entered. DMA access is available to appropriately configured L1 memories.

In the active mode, it is possible to disable the PLL through the PLL control register (PLL_CTL). If disabled, the PLL must be re-enabled before transitioning to the full-on or sleep modes.

**Table 4. Power Settings**

| Mode | PLL | PLL Bypassed | Core Clock (CCLK) | System Clock (SCLK) | Internal Power (VDDINT) |
|---|---|---|---|---|---|
| Full On | Enabled | No | Enabled | Enabled | On |
| Active | Enabled/ Disabled | Yes | Enabled | Enabled | On |
| Sleep | Enabled | — | Disabled | Enabled | On |
| Deep Sleep | Disabled | — | Disabled | Disabled | On |
| Hibernate | Disabled | — | Disabled | Disabled | Off |

### *Sleep Operating Mode—High Dynamic Power Savings*

The sleep mode reduces dynamic power dissipation by disabling the clock to the processor core (CCLK). The PLL and system clock (SCLK), however, continue to operate in this mode. Typically an external event or RTC activity wakes up the processor. When in the sleep mode, asserting wakeup causes the processor to sense the value of the BYPASS bit in the PLL control register (PLL_CTL). If BYPASS is disabled, the processor transitions to the full on mode. If BYPASS is enabled, the processor transitions to the active mode.

System DMA access to L1 memory is not supported in sleep mode.

### *Deep Sleep Operating Mode—Maximum Dynamic Power Savings*

The deep sleep mode maximizes dynamic power savings by disabling the clocks to the processor core (CCLK) and to all synchronous peripherals (SCLK). Asynchronous peripherals, such as the RTC, may still be running but cannot access internal resources or external memory. This powered-down mode can only be exited by assertion of the reset interrupt ($\overline{\text{RESET}}$) or by an asynchronous interrupt generated by the RTC. When in deep sleep mode, an RTC asynchronous interrupt causes the processor to transition to the active mode. Assertion of $\overline{\text{RESET}}$ while in deep sleep mode causes the processor to transition to the full-on mode.

### *Hibernate Operating Mode—Maximum Static Power Savings*

The hibernate mode maximizes static power savings by disabling the voltage and clocks to the processor core (CCLK) and to all of the synchronous peripherals (SCLK). The internal voltage regulator for the processor can be shut off by writing b#00 to the FREQ bits of the VR_CTL register. This disables both CCLK and SCLK. Furthermore, it sets the internal power supply voltage ($V_{DDINT}$) to 0 V to provide the greatest power savings. To preserve the processor state, prior to removing power, any critical information stored internally (memory contents, register contents, etc.) must be written to a non volatile storage device.

Since $V_{DDEXT}$ is still supplied in this mode, all of the external pins three-state, unless otherwise specified. This allows other devices that are connected to the processor to still have power applied without drawing unwanted current.

The Ethernet or CAN modules can wake up the internal supply regulator. If the PH6 pin is does not connect as $\overline{\text{PHYINT}}$ signal to an external PHY device it can be pulled low by any other device to wake the processor up. The regulator can also be woken up by a real-time clock wakeup event or by asserting the $\overline{\text{RESET}}$ pin, both of which initiate the hardware reset sequence. Individual sources are enabled by the VR_CTL register.

With the exception of the VR_CTL and the RTC registers, all internal registers and memories lose their content in the hibernate state. State variables may be held in external SRAM or SDRAM. The CKELOW bit in the VR_CTL register controls whether SDRAM operates in self-refresh mode which allows it to retain its content while the processor is in reset.

### *Power Savings*

As shown in Table 5, the processors support three different power domains which maximizes flexibility, while maintaining compliance with industry standards and conventions. By isolating the internal logic of the processor into its own power domain, separate from the RTC and other I/O, the processor can take advantage of dynamic power management, without affecting the RTC or other I/O devices. There are no sequencing requirements for the various power domains.

**Table 5. Power Domains**

| Power Domain | $V_{DD}$ Range |
|---|---|
| All internal logic, except RTC | $V_{DDINT}$ |
| RTC internal logic and crystal I/O | $V_{DDRTC}$ |
| All other I/O | $V_{DDEXT}$ |

The dynamic power management feature allows both the processor's input voltage ($V_{DDINT}$) and clock frequency ($f_{CCLK}$) to be dynamically controlled.

The power dissipated by a processor is largely a function of its clock frequency and the square of the operating voltage. For example, reducing the clock frequency by 25% results in a 25% reduction in power dissipation, while reducing the voltage by 25% reduces power dissipation by more than 40%. Further, these power savings are additive, in that if the clock frequency and supply voltage are both reduced, the power savings can be dramatic, as shown in the following equations.

The power savings factor is calculated as:

$$\text{power savings factor} = \frac{f_{CCLKRED}}{f_{CCLKNOM}} \times \left(\frac{V_{DDINTRED}}{V_{DDINTNOM}}\right)^2 \times \left(\frac{T_{RED}}{T_{NOM}}\right)$$

where the variables in the equations are:

$f_{CCLKNOM}$ is the nominal core clock frequency

$f_{CCLKRED}$ is the reduced core clock frequency

$V_{DDINTNOM}$ is the nominal internal supply voltage

$V_{DDINTRED}$ is the reduced internal supply voltage

$T_{NOM}$ is the duration running at $f_{CCLKNOM}$

$T_{RED}$ is the duration running at $f_{CCLKRED}$

The percent power savings is calculated as:

$$\%\ \text{power savings} = (1 - \text{power savings factor}) \times 100\%$$

## VOLTAGE REGULATION

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processor provides an on-chip voltage regulator that can generate processor core voltage levels (0.85 V to 1.2 V guaranteed from –5% to +10%) from an external 2.25 V to 3.6 V supply. Figure 5 shows the typical external components required to complete the power management system. The regulator controls the internal logic voltage levels and is programmable with the voltage regulator control register (VR_CTL) in increments of 50 mV. To reduce standby power consumption, the internal voltage regulator can be programmed to remove power to the processor core while keeping I/O power supplied. While in hibernate mode, $V_{DDEXT}$ can still be applied, eliminating the need for external buffers. The voltage regulator can be activated from this power-down state by asserting the $\overline{\text{RESET}}$ pin, which then initiates a boot sequence. The regulator can also be disabled and bypassed at the user's discretion. For additional information on voltage regulation, see "Switching Regulator Design Considerations for the ADSP-BF533 Blackfin Processors" (EE-228).

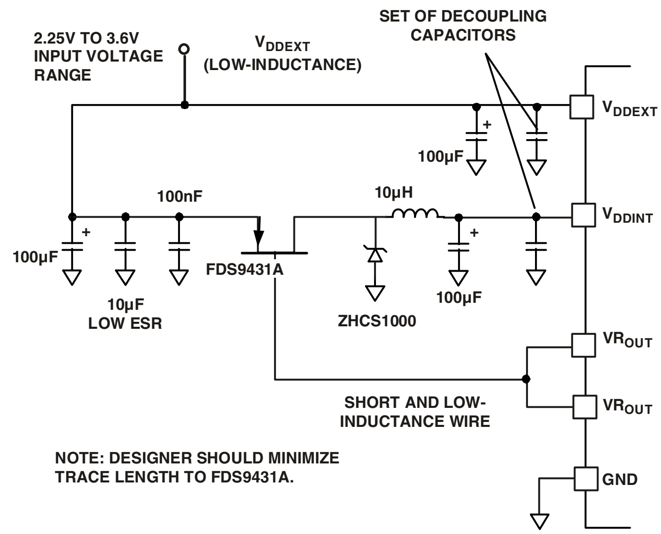



*Figure 5. Voltage Regulator Circuit*

## CLOCK SIGNALS

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processor can be clocked by an external crystal, a sine wave input, or a buffered, shaped clock derived from an external clock oscillator.

If an external clock is used, it should be a TTL compatible signal and must not be halted, changed, or operated below the specified frequency during normal operation. This signal is connected to the processor's CLKIN pin. When an external clock is used, the XTAL pin must be left unconnected.

Alternatively, because the processors include an on-chip oscillator circuit, an external crystal may be used. For fundamental frequency operation, use the circuit shown in Figure 6. A parallel-resonant, fundamental frequency, microprocessor-grade crystal is connected across the CLKIN and XTAL pins. The on-chip resistance between CLKIN and the XTAL pin is in the 500 kΩ range. Further parallel resistors are typically not recommended. The two capacitors and the series resistor shown in Figure 6 fine-tune phase and amplitude of the sine frequency.

The capacitor and resistor values shown in Figure 6 are typical values only. The capacitor values are dependent upon the crystal manufacturers' load capacitance recommendations and the PCB physical layout. The resistor value depends on the drive level specified by the crystal manufacturer. The user should verify the customized values based on careful investigations of multiple devices over temperature range.

A third-overtone crystal can be used for frequencies above 25 MHz. The circuit is then modified to ensure crystal operation only at the third overtone, by adding a tuned inductor circuit as shown in Figure 6. A design procedure for third-overtone operation is discussed in detail in the application note "Using Third Overtone Crystals with the ADSP-218x DSP" (EE-168).

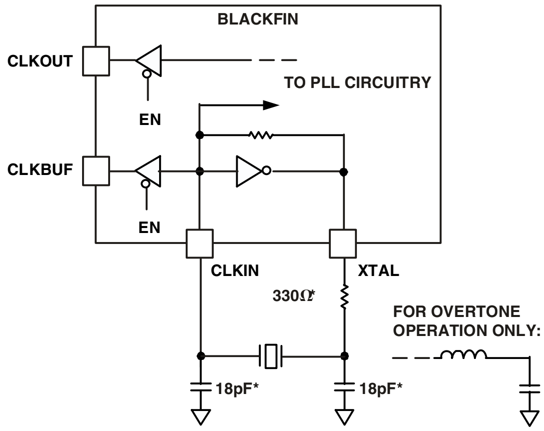



*Figure 6. External Crystal Connections*

The CLKBUF pin is an output pin, and is a buffer version of the input clock. This pin is particularly useful in Ethernet applications to limit the number of required clock sources in the system. In this type of application, a single 25 MHz or 50 MHz crystal may be applied directly to the processors. The 25 MHz or 50 MHz output of CLKBUF can then be connected to an external Ethernet MII or RMII PHY device.

Because of the default 10× PLL multiplier, providing a 50 MHz CLKIN exceeds the recommended operating conditions of the lower speed grades. Because of this restriction, an RMII PHY requiring a 50 MHz clock input cannot be clocked directly from the CLKBUF pin for the lower speed grades. In this case, either provide a separate 50 MHz clock source, or use an RMII PHY with 25 MHz clock input options. The CLKBUF output is active by default and can be disabled using the VR_CTL register for power savings.

The Blackfin core runs at a different clock rate than the on-chip peripherals. As shown in Figure 7, the core clock (CCLK) and system peripheral clock (SCLK) are derived from the input clock (CLKIN) signal. An on-chip PLL is capable of multiplying the CLKIN signal by a programmable 0.5× to 64× multiplication factor (bounded by specified minimum and maximum VCO frequencies). The default multiplier is 10×, but it can be modified by a software instruction sequence in the PLL_CTL register.

On-the-fly CCLK and SCLK frequency changes can be effected by simply writing to the PLL_DIV register. Whereas the maximum allowed CCLK and SCLK rates depend on the applied voltages $V_{DDINT}$ and $V_{DDEXT}$, the VCO is always permitted to run up to the frequency specified by the part's speed grade. The CLKOUT pin reflects the SCLK frequency to the off-chip world. It belongs to the SDRAM interface, but it functions as reference signal in other timing specifications as well. While active by default, it can be disabled using the EBIU_SDGCTL and EBIU_AMGCTL registers.

All on-chip peripherals are clocked by the system clock (SCLK). The system clock frequency is programmable by means of the SSEL3–0 bits of the PLL_DIV register. The values programmed into the SSEL fields define a divide ratio between the PLL output (VCO) and the system clock. SCLK divider values are 1 through 15. Table 6 illustrates typical system clock ratios.

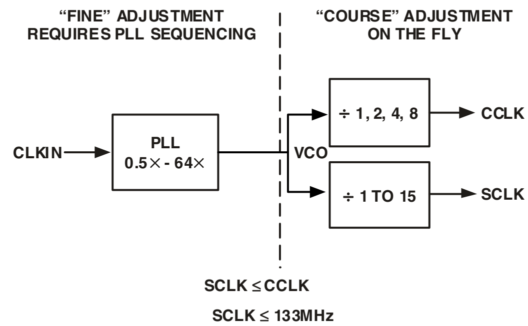



*Figure 7. Frequency Modification Methods*

Note that the divisor ratio must be chosen to limit the system clock frequency to its maximum of $f_{SCLK}$. The SSEL value can be changed dynamically without any PLL lock latencies by writing the appropriate values to the PLL divisor register (PLL_DIV).

**Table 6. Example System Clock Ratios**

| Signal Name SSEL3–0 | Divider Ratio VCO/SCLK | Example Frequency Ratios (MHz) | |
|---|---|---|---|
| | | VCO | SCLK |
| 0001 | 1:1 | 100 | 100 |
| 0110 | 6:1 | 300 | 50 |
| 1010 | 10:1 | 500 | 50 |

The core clock (CCLK) frequency can also be dynamically changed by means of the CSEL1–0 bits of the PLL_DIV register. Supported CCLK divider ratios are 1, 2, 4, and 8, as shown in Table 7. This programmable core clock capability is useful for fast core frequency modifications.

**Table 7. Core Clock Ratios**

| Signal Name CSEL1–0 | Divider Ratio VCO/CCLK | Example Frequency Ratios (MHz) | |
|---|---|---|---|
| | | VCO | CCLK |
| 00 | 1:1 | 300 | 300 |
| 01 | 2:1 | 300 | 150 |
| 10 | 4:1 | 500 | 125 |
| 11 | 8:1 | 200 | 25 |

The maximum CCLK frequency not only depends on the part's speed grade (see Ordering Guide on Page 67), it also depends on the applied $V_{DDINT}$ voltage. See Table 12, Table 13, and Table 14 on Page 27 for details. The maximal system clock rate (SCLK) depends on the chip package and the applied $V_{DDEXT}$ voltage (see Table 16 on Page 28).

## BOOTING MODES

The ADSP-BF534/ADSP-BF536/ADSP-BF537 processor has six mechanisms (listed in Table 8) for automatically loading internal and external memory after a reset. A seventh mode is provided to execute from external memory, bypassing the boot sequence.

**Table 8. Booting Modes**

| BMODE2–0 | Description |
|---|---|
| 000 | Execute from 16-bit external memory (bypass boot ROM) |
| 001 | Boot from 8-bit or 16-bit memory (EPROM/flash) |
| 010 | Reserved |
| 011 | Boot from serial SPI memory (EEPROM/flash) |
| 100 | Boot from SPI host (slave mode) |
| 101 | Boot from serial TWI memory (EEPROM/flash) |
| 110 | Boot from TWI host (slave mode) |
| 111 | Boot from UART host (slave mode) |

The BMODE pins of the reset configuration register, sampled during power-on resets and software-initiated resets, implement the following modes:

- Execute from 16-bit external memory – Execution starts from address 0x2000 0000 with 16-bit packing. The boot ROM is bypassed in this mode. All configuration settings are set for the slowest device possible (3-cycle hold time; 15-cycle R/W access times; 4-cycle setup).
- Boot from 8-bit and 16-bit external flash memory – The 8-bit or 16-bit flash boot routine located in Boot ROM memory space is set up using asynchronous memory bank 0. All configuration settings are set for the slowest device possible (3-cycle hold time; 15-cycle R/W access times; 4-cycle setup). The Boot ROM evaluates the first byte of the boot stream at address 0x2000 0000. If it is 0x40, 8-bit boot is performed. A 0x60 byte assumes a 16-bit memory device and performs 8-bit DMA. A 0x20 byte also assumes 16-bit memory but performs 16-bit DMA.
- Boot from serial SPI memory (EEPROM or flash) – 8-, 16-, or 24-bit addressable devices are supported as well as AT45DB041, AT45DB081, AT45DB161, AT45DB321, AT45DB642, and AT45DB1282 DataFlash® devices from Atmel. The SPI uses the PF10/SPI SSEL1 output pin to select a single SPI EEPROM/flash device, submits a read command and successive address bytes (0x00) until a valid 8-, 16-, or 24-bit, or Atmel addressable device is detected, and begins clocking data into the processor.
- Boot from SPI host device – The Blackfin processor operates in SPI slave mode and is configured to receive the bytes of the .LDR file from an SPI host (master) agent. To hold off the host device from transmitting while the boot ROM is busy, the Blackfin processor asserts a GPIO pin, called host wait (HWAIT), to signal the host device not to send any more bytes until the flag is deasserted. The flag is chosen by the user and this information is transferred to the Blackfin processor via bits 10:5 of the FLAG header.
- Boot from UART – Using an autobaud handshake sequence, a boot-stream-formatted program is downloaded by the host. The host agent selects a baud rate within the UART's clocking capabilities. When performing the autobaud, the UART expects an "@" (boot stream) character (8 bits data, 1 start bit, 1 stop bit, no parity bit) on the RXD pin to determine the bit rate. It then replies with an acknowledgement that is composed of 4 bytes: 0xBF, the value of UART_DLL, the value of UART_DLH, and 0x00. The host can then download the boot stream. When the processor needs to hold off the host, it deasserts CTS. Therefore, the host must monitor this signal.
- Boot from serial TWI memory (EEPROM/flash) – The Blackfin processor operates in master mode and selects the TWI slave with the unique ID 0xA0. It submits successive read commands to the memory device starting at two byte internal address 0x0000 and begins clocking data into the processor. The TWI memory device should comply with Philips I$^2$C Bus Specification version 2.1 and have the capability to auto-increment its internal address counter such that the contents of the memory device can be read sequentially.
- Boot from TWI host – The TWI host agent selects the slave with the unique ID 0x5F. The processor replies with an acknowledgement and the host can then download the boot stream. The TWI host agent should comply with Philips I$^2$C Bus Specification version 2.1. An I$^2$C multiplexer can be used to select one processor at a time when booting multiple processors from a single TWI.

For each of the boot modes, a 10-byte header is first brought in from an external device. The header specifies the number of bytes to be transferred and the memory destination address. Multiple memory blocks may be loaded by any boot sequence. Once all blocks are loaded, program execution commences from the start of L1 instruction SRAM.

In addition, Bit 4 of the reset configuration register can be set by application code to bypass the normal boot sequence during a software reset. For this case, the processor jumps directly to the beginning of L1 instruction memory.

To augment the boot modes, a secondary software loader can be added to provide additional booting mechanisms. This secondary loader could provide the capability to boot from flash, variable baud rate, and other sources. In all boot modes except bypass, program execution starts from on-chip L1 memory address 0xFFA0 0000.

## INSTRUCTION SET DESCRIPTION

The Blackfin processor family assembly language instruction set employs an algebraic syntax designed for ease of coding and readability. The instructions have been specifically tuned to provide a flexible, densely encoded instruction set that compiles to a very small final memory size. The instruction set also provides fully featured multifunction instructions that allow the

programmer to use many of the processor core resources in a single instruction. Coupled with many features more often seen on microcontrollers, this instruction set is very efficient when compiling C and C++ source code. In addition, the architecture supports both user (algorithm/application code) and supervisor (O/S kernel, device drivers, debuggers, ISRs) modes of operation, allowing multiple levels of access to core processor resources.

The assembly language, which takes advantage of the processor's unique architecture, offers the following advantages:

- Seamlessly integrated DSP/MCU features are optimized for both 8-bit and 16-bit operations.
- A multi-issue load/store modified-Harvard architecture, which supports two 16-bit MAC or four 8-bit ALU + two load/store + two pointer updates per cycle.
- All registers, I/O, and memory are mapped into a unified 4G byte memory space, providing a simplified programming model.
- Microcontroller features, such as arbitrary bit and bit-field manipulation, insertion, and extraction; integer operations on 8-, 16-, and 32-bit data-types; and separate user and supervisor stack pointers.
- Code density enhancements, which include intermixing of 16-bit and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded in 16 bits.

## DEVELOPMENT TOOLS

The Blackfin is supported with a complete set of CROSSCORE®† software and hardware development tools, including Analog Devices emulators and the VisualDSP++®‡ development environment. The same emulator hardware that supports other Analog Devices processors also fully emulates the Blackfin.

The VisualDSP++ project management environment lets programmers develop and debug an application. This environment includes an easy to use assembler that is based on an algebraic syntax, an archiver (librarian/library builder), a linker, a loader, a cycle-accurate instruction-level simulator, a C/C++ compiler, and a C/C++ runtime library that includes DSP and mathematical functions. A key point for these tools is C/C++ code efficiency. The compiler has been developed for efficient translation of C/C++ code to Blackfin assembly. The Blackfin processor has architectural features that improve the efficiency of compiled C/C++ code.

The VisualDSP++ debugger has a number of important features. Data visualization is enhanced by a plotting package that offers a significant level of flexibility. This graphical representation of user data enables the programmer to quickly determine the performance of an algorithm. As algorithms grow in complexity, this capability can have increasing significance on the designer's development schedule, increasing productivity. Statistical profiling enables the programmer to nonintrusively poll the processor as it is running the program. This feature, unique to VisualDSP++, enables the software developer to passively gather important code execution metrics without interrupting the real-time characteristics of the program. Essentially, the developer can identify bottlenecks in software quickly and efficiently. By using the profiler, the programmer can focus on those areas in the program that impact performance and take corrective action.

Debugging both C/C++ and assembly programs with the VisualDSP++ debugger, programmers can:

- View mixed C/C++ and assembly code (interleaved source and object information).
- Insert breakpoints.
- Set conditional breakpoints on registers, memory, and stacks.
- Trace instruction execution.
- Perform linear or statistical profiling of program execution.
- Fill, dump, and graphically plot the contents of memory.
- Perform source level debugging.
- Create custom debugger windows.

The VisualDSP++ IDE lets programmers define and manage software development. Its dialog boxes and property pages let programmers configure and manage all development tools, including color syntax highlighting in the VisualDSP++ editor. These capabilities permit programmers to:

- Control how the development tools process inputs and generate outputs.
- Maintain a one-to-one correspondence with the tool's command line switches.

The VisualDSP++ Kernel (VDK) incorporates scheduling and resource management tailored specifically to address the memory and timing constraints of embedded, real-time programming. These capabilities enable engineers to develop code more effectively, eliminating the need to start from the very beginning when developing new application code. The VDK features include threads, critical and unscheduled regions, semaphores, events, and device flags. The VDK also supports priority-based, pre-emptive, cooperative, and time-sliced scheduling approaches. In addition, the VDK was designed to be scalable. If the application does not use a specific feature, the support code for that feature is excluded from the target system.

Because the VDK is a library, a developer can decide whether to use it or not. The VDK is integrated into the VisualDSP++ development environment, but can also be used with standard command line tools. When the VDK is used, the development environment assists the developer with many error prone tasks and assists in managing system resources, automating the generation of various VDK-based objects, and visualizing the system state when debugging an application that uses the VDK.

† CROSSCORE is a registered trademark of Analog Devices, Inc.
‡ VisualDSP++ is a registered trademark of Analog Devices, Inc.

VCSE is Analog Devices' technology for creating, using, and reusing software components (independent modules of substantial functionality) to quickly and reliably assemble software applications. Components can be downloaded from the Web and dropped into the application. Component archives can be published from within VisualDSP++. VCSE supports component implementation in C/C++ or assembly language.

The expert linker can be used to visually manipulate the placement of code and data in the embedded system. Memory utilization can be viewed in a color-coded graphical form. Code and data can be easily moved to different areas of the processor or external memory with the drag of the mouse. Runtime stack and heap usage can be examined. The expert linker is fully compatible with existing linker definition file (LDF), allowing the developer to move between the graphical and textual environments.

Analog Devices emulators use the IEEE 1149.1 JTAG test access port of the Blackfin to monitor and control the target board processor during emulation. The emulator provides full-speed emulation, allowing inspection and modification of memory, registers, and processor stacks. Nonintrusive in-circuit emulation is assured by the use of the processor's JTAG interface—the emulator does not affect target system loading or timing.

In addition to the software and hardware development tools available from Analog Devices, third parties provide a wide range of tools supporting the Blackfin processor family. Third party software tools include DSP libraries, real-time operating systems, and block diagram design tools.

### *EZ-KIT Lite® Evaluation Board*

For evaluation of ADSP-BF534/ADSP-BF536/ADSP-BF537 processors, use the ADSP-BF537 EZ-KIT Lite board available from Analog Devices. Order part number ADDS-BF537-EZLITE. The board comes with on-chip emulation capabilities and is equipped to enable software development. Multiple daughter cards are available.

## DESIGNING AN EMULATOR-COMPATIBLE PROCESSOR BOARD

The Analog Devices family of emulators are tools that every system developer needs in order to test and debug hardware and software systems. Analog Devices has supplied an IEEE 1149.1 JTAG Test Access Port (TAP) on each JTAG processor. The emulator uses the TAP to access the internal features of the processor, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The processor must be halted to send data and commands, but once an operation has been completed by the emulator, the processor system is set running at full speed with no impact on system timing.

To use these emulators, the target board must include a header that connects the processor's JTAG port to the emulator.

For details on target board design issues including mechanical layout, single processor connections, multiprocessor scan chains, signal buffering, signal termination, and emulator pod logic, see "Analog Devices JTAG Emulation Technical Reference" (EE-68) on the Analog Devices website under www.analog.com/ee-notes. This document is updated regularly to keep pace with improvements to emulator support.

## RELATED DOCUMENTS

The following publications that describe the ADSP-BF534/ADSP-BF536/ADSP-BF537 processors (and related processors) can be ordered from any Analog Devices sales office or accessed electronically on our website:

- *Getting Started with Blackfin Processors*
- *ADSP-BF537 Blackfin Processor Hardware Reference*
- *ADSP-BF53x/ADSP-BF56x Blackfin Processor Programming Reference*
- *ADSP-BF537 Blackfin Processor Anomaly List*

# PIN DESCRIPTIONS

ADSP-BF534/ADSP-BF536/ADSP-BF537 processor's pin definitions are listed in Table 9. In order to maintain maximum function and reduce package size and pin count, some pins have dual, multiplexed functions. In cases where pin function is reconfigurable, the default state is shown in plain text, while the alternate function is shown in italics. Pins shown with an asterisk after their name (*) offer high source/high sink current capabilities.

All pins are three-stated during and immediately after reset, with the exception of the external memory interface and the buffered XTAL output pin (CLKBUF). On the external memory interface, the control and address lines are driven high during reset unless the $\overline{\text{BR}}$ pin is asserted.

All I/O pins have their input buffers disabled with the exception of the pins noted in the data sheet that need pull-ups or pull-downs if unused.

The SDA (serial data) and SCL (serial clock) pins are open drain and therefore require a pull-up resistor. Consult version 2.1 of the $I^2C$ specification for the proper resistor value.

**Table 9. Pin Descriptions**

| Pin Name | Type | Function | Driver Type[1] | Pull-Up/Pull-Down |
|---|---|---|---|---|
| *Memory Interface* | | | | |
| ADDR19–1 | O | Address Bus for Async Access | A | |
| DATA15–0 | I/O | Data Bus for Async/Sync Access | A | |
| $\overline{\text{ABE1–0}}$/SDQM1–0 | O | Byte Enables/Data Masks for Async/Sync Access | A | |
| $\overline{\text{BR}}$ | I | Bus Request | | This pin should be pulled high when not used |
| $\overline{\text{BG}}$ | O | Bus Grant | A | |
| $\overline{\text{BGH}}$ | O | Bus Grant Hang | A | |
| *Asynchronous Memory Control* | | | | |
| $\overline{\text{AMS3–0}}$ | O | Bank Select | A | |
| ARDY | I | Hardware Ready Control | | |
| $\overline{\text{AOE}}$ | O | Output Enable | A | |
| $\overline{\text{ARE}}$ | O | Read Enable | A | |
| $\overline{\text{AWE}}$ | O | Write Enable | A | |
| *Synchronous Memory Control* | | | | |
| $\overline{\text{SRAS}}$ | O | Row Address Strobe | A | |
| $\overline{\text{SCAS}}$ | O | Column Address Strobe | A | |
| $\overline{\text{SWE}}$ | O | Write Enable | A | |
| SCKE | O | Clock Enable | A | SCKE is three-stated during hibernate. A pull-down resistor is required on SCKE in conjunction with setting the CKELOW bit to enable self-refresh during hibernate. |
| CLKOUT | O | Clock Output | B | |
| SA10 | O | A10 Pin | A | |
| $\overline{\text{SMS}}$ | O | Bank Select | A | |

**Table 9. Pin Descriptions (Continued)**

| Pin Name | Type | Function | Driver Type[1] | Pull-Up/Pull-Down |
|---|---|---|---|---|
| *Port F: GPIO/UART1–0/Timer7–0/SPI/ External DMA Request (* = High Source/High Sink Pin)* | | | | |
| PF0* – GPIO/*UART0 TX/DMAR0* | I/O | GPIO/*UART0 Transmit/DMA Request 0* | C | |
| PF1* – GPIO/*UART0 RX/DMAR1/TACI1* | I/O | GPIO/*UART0 Receive/DMA Request 1/Timer1 Alternate Input Capture* | C | |
| PF2* – GPIO/*UART1 TX/TMR7* | I/O | GPIO/*UART1 Transmit/Timer7* | C | |
| PF3* – GPIO/*UART1 RX/TMR6/TACI6* | I/O | GPIO/*UART1 Receive/Timer6/Timer6 Alternate Input Capture* | C | |
| PF4* – GPIO/*TMR5/SPI SSEL6* | I/O | GPIO/*Timer5/SPI Slave Select Enable 6* | C | |
| PF5* – GPIO/*TMR4/SPI SSEL5* | I/O | GPIO/*Timer4/SPI Slave Select Enable 5* | C | |
| PF6* – GPIO/*TMR3/SPI SSEL4* | I/O | GPIO/*Timer3/SPI Slave Select Enable 4* | C | |
| PF7* – GPIO/*TMR2/PPI FS3* | I/O | GPIO/*Timer2/PPI Frame Sync 3* | C | |
| PF8 – GPIO/*TMR1/PPI FS2* | I/O | GPIO/*Timer1/PPI Frame Sync 2* | C | |
| PF9 – GPIO/*TMR0/PPI FS1* | I/O | GPIO/*Timer0/PPI Frame Sync 1* | C | |
| PF10 – GPIO/*SPI SSEL1* | I/O | GPIO/*SPI Slave Select Enable 1* | C | |
| PF11 – GPIO/*SPI MOSI* | I/O | GPIO/*SPI Master Out Slave In* | C | |
| PF12 – GPIO/*SPI MISO* | I/O | GPIO/*SPI Master In Slave Out* | C | This pin should always be pulled high through a 4.7 kΩ resistor if booting via the SPI port |
| PF13 – GPIO/*SPI SCK* | I/O | GPIO/*SPI Clock* | D | |
| PF14 – GPIO/*SPI SS/TACLK0* | I/O | GPIO/*SPI Slave Select/Alternate Timer0 Clock Input* | C | |
| PF15 – GPIO/*PPI CLK/TMRCLK* | I/O | GPIO/*PPI Clock/External Timer Reference* | C | |
| *Port G: GPIO/PPI/SPORT1* | | | | |
| PG0 – GPIO/*PPI D0* | I/O | GPIO/*PPI Data 0* | C | |
| PG1 – GPIO/*PPI D1* | I/O | GPIO/*PPI Data 1* | C | |
| PG2 – GPIO/*PPI D2* | I/O | GPIO/*PPI Data 2* | C | |
| PG3 – GPIO/*PPI D3* | I/O | GPIO/*PPI Data 3* | C | |
| PG4 – GPIO/*PPI D4* | I/O | GPIO/*PPI Data 4* | C | |
| PG5 – GPIO/*PPI D5* | I/O | GPIO/*PPI Data 5* | C | |
| PG6 – GPIO/*PPI D6* | I/O | GPIO/*PPI Data 6* | C | |
| PG7 – GPIO/*PPI D7* | I/O | GPIO/*PPI Data 7* | C | |
| PG8 – GPIO/*PPI D8/DR1SEC* | I/O | GPIO/*PPI Data 8/SPORT1 Receive Data Secondary* | C | |
| PG9 – GPIO/*PPI D9/DT1SEC* | I/O | GPIO/*PPI Data 9/SPORT1 Transmit Data Secondary* | C | |
| PG10 – GPIO/*PPI D10/RSCLK1* | I/O | GPIO/*PPI Data 10/SPORT1 Receive Serial Clock* | D | |
| PG11 – GPIO/*PPI D11/RFS1* | I/O | GPIO/*PPI Data 11/SPORT1 Receive Frame Sync* | C | |
| PG12 – GPIO/*PPI D12/DR1PRI* | I/O | GPIO/*PPI Data 12/SPORT1 Receive Data Primary* | C | |
| PG13 – GPIO/*PPI D13/TSCLK1* | I/O | GPIO/*PPI Data 13/SPORT1 Transmit Serial Clock* | D | |
| PG14 – GPIO/*PPI D14/TFS1* | I/O | GPIO/*PPI Data 14/SPORT1 Transmit Frame Sync* | C | |
| PG15 – GPIO/*PPI D15/DT1PRI* | I/O | GPIO/*PPI Data 15/SPORT1 Transmit Data Primary* | C | |

**Table 9. Pin Descriptions (Continued)**

| Pin Name | Type | Function | Driver Type[1] | Pull-Up/Pull-Down |
|---|---|---|---|---|
| *Port H: GPIO/10/100 Ethernet MAC* (On ADSP-BF534, these pins are GPIO only) | | | | |
| PH0 – GPIO/*ETxD0* | I/O | GPIO/*Ethernet MII or RMII Transmit D0* | E | |
| PH1 – GPIO/*ETxD1* | I/O | GPIO/*Ethernet MII or RMII Transmit D1* | E | |
| PH2 – GPIO/*ETxD2* | I/O | GPIO/*Ethernet MII Transmit D2* | E | |
| PH3 – GPIO/*ETxD3* | I/O | GPIO/*Ethernet MII Transmit D3* | E | |
| PH4 – GPIO/*ETxEN* | I/O | GPIO/*Ethernet MII or RMII Transmit Enable* | E | |
| PH5 – GPIO/*MII TxCLK/RMII REF_CLK* | I/O | GPIO/*Ethernet MII Transmit Clock/RMII Reference Clock* | E | |
| PH6 – GPIO/*MII* $\overline{PHYINT}$*/RMII* $\overline{MDINT}$ | I/O | GPIO/*Ethernet MII PHY Interrupt/RMII Management Data Interrupt* | E | When driven low PH6 pin can be used to wakeup the processor from hibernate state, regardless whether used in Ethernet mode as $\overline{PHYINT}$ or in normal GPIO mode. If used for wake up, pull the signal up by a resistor and enable the feature by the PHYWE bit in the VR_CTL register. |
| PH7 – GPIO/*COL* | I/O | GPIO/*Ethernet Collision* | E | |
| PH8 – GPIO/*ERxD0* | I/O | GPIO/*Ethernet MII or RMII Receive D0* | E | |
| PH9 – GPIO/*ERxD1* | I/O | GPIO/*Ethernet MII or RMII Receive D1* | E | |
| PH10 – GPIO/*ERxD2* | I/O | GPIO/*Ethernet MII Receive D2* | E | |
| PH11 – GPIO/*ERxD3* | I/O | GPIO/*Ethernet MII Receive D3* | E | |
| PH12 – GPIO/*ERxDV/TACLK5* | I/O | GPIO/*Ethernet MII Receive Data Valid/Alternate Timer5 Input Clock* | E | |
| PH13 – GPIO/*ERxCLK/TACLK6* | I/O | GPIO/*Ethernet MII Receive Clock/Alternate Timer6 Input Clock* | E | |
| PH14 – GPIO/*ERxER/TACLK7* | I/O | GPIO/*Ethernet MII or RMII Receive Error/Alternate Timer7 Input Clock* | E | |
| PH15 – GPIO/*MII CRS/RMII CRS_DV* | I/O | GPIO/*Ethernet MII Carrier Sense/Ethernet RMII Carrier Sense and Receive Data Valid* | E | |

Table 9. Pin Descriptions (Continued)

| Pin Name | Type | Function | Driver Type[1] | Pull-Up/Pull-Down |
|---|---|---|---|---|
| *Port J: SPORT0/TWI/SPI Select/CAN* | | | | |
| PJ0 – MDC | O | Ethernet Management Channel Clock | E | On ADSP-BF534 processors, do not connect PJ0, and tie PJ1 to ground |
| PJ1 – MDIO | I/O | Ethernet Management Channel Serial Data | E | On ADSP-BF534 processors, do not connect PJ0, and tie PJ1 to ground |
| PJ2 – SCL | I/O | TWI Serial Clock | F | |
| PJ3 – SDA | I/O | TWI Serial Data | F | |
| PJ4 – DR0SEC/*CANRX*/*TACI0* | I | SPORT0 Receive Data Secondary/*CAN Receive*/*Timer0 Alternate Input Capture* | | |
| PJ5 – DT0SEC/*CANTX*/*SPI SSEL7* | O | SPORT0 Transmit Data Secondary/*CAN Transmit*/*SPI Slave Select Enable 7* | C | |
| PJ6 – RSCLK0/*TACLK2* | I/O | SPORT0 Receive Serial Clock/*Alternate Timer2 Clock Input* | D | |
| PJ7 – RFS0/*TACLK3* | I/O | SPORT0 Receive Frame Sync/*Alternate Timer3 Clock Input* | C | |
| PJ8 – DR0PRI/*TACLK4* | I | SPORT0 Receive Data Primary/*Alternate Timer4 Clock Input* | | |
| PJ9 – TSCLK0/*TACLK1* | I/O | SPORT0 Transmit Serial Clock/*Alternate Timer1 Clock Input* | D | |
| PJ10 – TFS0/*SPI SSEL3* | I/O | SPORT0 Transmit Frame Sync/*SPI Slave Select Enable 3* | C | |
| PJ11 – DT0PRI/*SPI SSEL2* | O | SPORT0 Transmit Data Primary/*SPI Slave Select Enable 2* | C | |
| *Real Time Clock* | | | | |
| RTXI | I | RTC Crystal Input | | This pin should always be pulled low when not used |
| RTXO | O | RTC Crystal Output | | |
| *JTAG Port* | | | | |
| TCK | I | JTAG Clock | | |
| TDO | O | JTAG Serial Data Out | C | |
| TDI | I | JTAG Serial Data In | | |
| TMS | I | JTAG Mode Select | | |
| $\overline{\text{TRST}}$ | I | JTAG Reset | | This pin should be pulled low if the JTAG port is not used |
| $\overline{\text{EMU}}$ | O | Emulation Output | C | |
| *Clock* | | | | |
| CLKIN | I | Clock/Crystal Input | | |
| XTAL | O | Crystal Output | | |
| CLKBUF | O | Buffered XTAL Output | E | E |
| *Mode Controls* | | | | |
| $\overline{\text{RESET}}$ | I | Reset | | |
| $\overline{\text{NMI}}$ | I | Nonmaskable Interrupt | | This pin should always be pulled high when not used |
| BMODE2–0 | I | Boot Mode Strap 2-0 | | |
| *Voltage Regulator* | | | | |
| VROUT0 | O | External FET Drive | | |
| VROUT1 | O | External FET Drive | | |

**Table 9. Pin Descriptions (Continued)**

| Pin Name | Type | Function | Driver Type[1] | Pull-Up/Pull-Down |
|---|---|---|---|---|
| *Supplies* | | | | |
| $V_{DDEXT}$ | P | I/O Power Supply | | |
| $V_{DDINT}$ | P | Internal Power Supply (regulated from 2.25 V to 3.6 V) | | |
| $V_{DDRTC}$ | P | Real Time Clock Power Supply | | |
| GND | G | External Ground | | |

[1] See Output Drive Currents on Page 51 for more information about each driver types.

# SPECIFICATIONS

Note that component specifications are subject to change without notice.

## OPERATING CONDITIONS

| Parameter | | | Min | Nominal | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DDINT}$ | Internal Supply Voltage[1, 2] | | 0.80 | 1.20 | 1.32 | V |
| $V_{DDINT}$ | Internal Supply Voltage[1, 3] | | 0.95 | 1.20 | 1.32 | V |
| $V_{DDEXT}$ | External Supply Voltage[2] | | 2.25 | 2.5 or 3.3 | 3.6 | V |
| $V_{DDEXT}$ | External Supply Voltage[3] | | 2.7 | 3.0 or 3.3 | 3.6 | V |
| $V_{DDRTC}$ | Real Time Clock Power Supply Voltage | | 2.25 | | 3.6 | V |
| $V_{IH}$ | High Level Input Voltage[4, 5] | @ $V_{DDEXT}$ = maximum | 2.0 | | 3.6 | V |
| $V_{IHCLKIN}$ | High Level Input Voltage[6] | @ $V_{DDEXT}$ = maximum | 2.2 | | 3.6 | V |
| $V_{IH5V}$ | 5.0 V Tolerant Pins, High Level Input Voltage[7] | @ $V_{DDEXT}$ = maximum | 2.0 | | 5.5 | V |
| $V_{IL}$ | Low Level Input Voltage[4, 8] | @ $V_{DDEXT}$ = minimum | –0.3 | | +0.6 | V |
| $V_{IL5V}$ | 5.0 V Tolerant Pins, Low Level Input Voltage[7] | @ $V_{DDEXT}$ = minimum | –0.3 | | +0.8 | V |
| $T_J$ | Junction Temperature | 208-Ball Chip Scale Ball Grid Array (Mini-BGA) @ $T_{AMBIENT}$ = –40°C to +105°C | –40 | | +120 | °C |
| $T_J$ | Junction Temperature | 208-Ball Chip Scale Ball Grid Array (Mini-BGA) @ $T_{AMBIENT}$ = –40°C to +85°C | –40 | | +105 | °C |
| $T_J$ | Junction Temperature | 182-Ball Chip Scale Ball Grid Array (Mini-BGA) @ $T_{AMBIENT}$ = –40°C to +85°C | –40 | | +105 | °C |

[1] The regulator can generate $V_{DDINT}$ at levels of 0.85 V to 1.2 V with –5 % to +10 % tolerance.
[2] Nonautomotive grade parts, see Ordering Guide on Page 67.
[3] Automotive grade parts, see Ordering Guide on Page 67.
[4] Bidirectional pins (DATA15–0, PF15–0, PG15–0, PH15–0, TFS0, TCLK0, RSCLK0, RFS0, MDIO) and input pins ($\overline{BR}$, ARDY, DR0PRI, DR0SEC, RTXI, TCK, TDI, TMS, $\overline{TRST}$, CLKIN, $\overline{RESET}$, $\overline{NMI}$, and BMODE2–0) of the ADSP-BF534/ADSP-BF536/ADSP-BF537 are 3.3 V-tolerant (always accept up to 3.6 V maximum $V_{IH}$). Voltage compliance (on outputs, $V_{OH}$) is limited by the $V_{DDEXT}$ supply voltage.
[5] Parameter value applies to all input and bidirectional pins except CLKIN, SDA, and SCL.
[6] Parameter value applies to CLKIN pin only.
[7] Pins SDA, SCL, and PJ4 are 5.0 V tolerant (always accept up to 5.5 V maximum $V_{IH}$). Voltage compliance (on outputs, $V_{OH}$) is limited by the $V_{DDEXT}$ supply voltage.
[8] Parameter value applies to all input and bidirectional pins except SDA and SCL.

## ELECTRICAL CHARACTERISTICS

| Parameter | Description | Test Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{OH}$ (All Outputs and I/Os Except Port F, Port G, Port H) | High Level Output Voltage[1] | @ $V_{DDEXT}$ = 3.3 V ± 10%, $I_{OH}$ = –0.5 mA<br>@ $V_{DDEXT}$ = 2.5 V ± 10%, $I_{OH}$ = –0.5 mA | $V_{DDEXT}$ – 0.5<br>$V_{DDEXT}$ – 0.5 | | V<br>V |
| $V_{OH}$ (Port F7–0) | | @ $V_{DDEXT}$ = 3.3 V ± 10%, $I_{OH}$ = –8 mA<br>@ $V_{DDEXT}$ = 2.5 V ± 10%, $I_{OH}$ = –6 mA | $V_{DDEXT}$ – 0.5<br>$V_{DDEXT}$ – 0.5 | | V<br>V |
| $V_{OH}$ (Port F15–8, Port G, Port H) | | $I_{OH}$ = –2 mA | $V_{DDEXT}$ – 0.5 | | V |
| $I_{OH}$ (Max Combined for Port F7–0) | | $V_{OH}$ = $V_{DDEXT}$ – 0.5 V min | | –64 | mA |
| $I_{OH}$ (Max Total for All Port F, Port G, and Port H Pins) | | $V_{OH}$ = $V_{DDEXT}$ – 0.5 V min | | –144 | mA |
| $V_{OL}$ (All Outputs and I/Os Except Port F, Port G, Port H) | Low Level Output Voltage[1] | @ $V_{DDEXT}$ = 3.3 V ± 10%, $I_{OL}$ = 2.0 mA<br>@ $V_{DDEXT}$ = 2.5 V ± 10%, $I_{OL}$ = 2.0 mA | | 0.4 | V |
| $V_{OL}$ (Port F7–0) | | @ $V_{DDEXT}$ = 3.3 V ± 10%, $I_{OL}$ = 8 mA<br>@ $V_{DDEXT}$ = 2.5 V ± 10%, $I_{OL}$ = 6 mA | | 0.5<br>0.5 | V<br>V |
| $V_{OL}$ (Port F15–8, Port G, Port H) | | $I_{OL}$ = 2 mA | | 0.5 | V |
| $I_{OL}$ (Max Combined for Port F7–0) | | $V_{OL}$ = 0.5 V max | | 64 | mA |
| $I_{OL}$ (Max Total for All Port F, Port G, and Port H Pins) | | $V_{OL}$ = 0.5 V max | | 144 | mA |
| $I_{IH}$ | High Level Input Current[2] | @ $V_{DDEXT}$ =3.6 V, $V_{IN}$ = 3.6 V | | 10 | μA |
| $I_{IH5V}$ | High Level Input Current[3] | @ $V_{DDEXT}$ =3.0 V, $V_{IN}$ = 5.5 V | | 10 | μA |
| $I_{IL}$ | Low Level Input Current[2] | @ $V_{DDEXT}$ =3.6 V, $V_{IN}$ = 0 V | | 10 | μA |
| $I_{IHP}$ | High Level Input Current JTAG[4] | @ $V_{DDEXT}$ = 3.6 V, $V_{IN}$ = 3.6 V | | 50.0 | μA |
| $I_{OZH}$ | Three-State Leakage Current[5] | @ $V_{DDEXT}$= 3.6 V, $V_{IN}$ = 3.6 V | | 10 | μA |
| $I_{OZH5V}$ | Three-State Leakage Current[6] | @ $V_{DDEXT}$ =3.0 V, $V_{IN}$ = 5.5 V | | 10 | μA |
| $I_{OZL}$ | Three-State Leakage Current[5] | @ $V_{DDEXT}$= 3.6 V, $V_{IN}$ = 0 V | | 10 | μA |
| $C_{IN}$ | Input Capacitance[7, 8] | $f_{IN}$ = 1 MHz, $T_{AMBIENT}$= 25°C, $V_{IN}$ = 2.5 V | | 8 | pF |

[1] Applies to output and bidirectional pins.
[2] Applies to input pins.
[3] Applies to input pin PJ4.
[4] Applies to JTAG input pins (TCK, TDI, TMS, $\overline{\text{TRST}}$).
[5] Applies to three-statable pins.
[6] Applies to bidirectional pins PJ2 and PJ3.
[7] Applies to all signal pins.
[8] Guaranteed, but not tested.

## ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed below may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions greater than those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

| Parameter | Rating |
|---|---|
| Internal (Core) Supply Voltage ($V_{DDINT}$) | −0.3 V to +1.4 V |
| External (I/O) Supply Voltage ($V_{DDEXT}$) | −0.3 V to +3.8 V |
| Input Voltage | −0.5 V to +3.6 V |
| Input Voltage[1] | −0.5 V to +5.5 V |
| Output Voltage Swing | −0.5 V to $V_{DDEXT}$ +0.5 V |
| Load Capacitance[2] | 200 pF |
| Storage Temperature Range | −65°C to +150°C |
| Junction Temperature Underbias | +125°C |

[1] Applies to 5 V tolerant pins SCL, SDA, and PJ4. For duty cycles, see Table 10.
[2] For proper SDRAM controller operation, the maximum load capacitance is 50 pF (at 3.3 V) or 30 pF (at 2.5 V) for ADDR19–1, DATA15–0, ABE1–0/SDQM1–0, CLKOUT, SCKE, SA10, SRAS, SCAS, SWE, and SMS.

**Table 10. Maximum Duty Cycle for Input[1] Transient Voltage**

| $V_{IN}$ Min (V) | $V_{IN}$ Max (V)[2] | Maximum Duty Cycle |
|---|---|---|
| –0.33 | 3.63 | 100% |
| –0.50 | 3.80 | 48% |
| –0.60 | 3.90 | 30% |
| –0.70 | 4.00 | 20% |
| –0.80 | 4.10 | 10% |
| –0.90 | 4.20 | 8% |
| –1.00 | 4.30 | 5% |

[1] Applies to all signal pins with the exception of CLKIN, XTAL, and VROUT1–0.
[2] Only one of the listed options can apply to a particular design.

## ESD SENSITIVITY

**ESD (electrostatic discharge) sensitive device.** Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be take to avoid performance degradation or loss of functionality.

## PACKAGE INFORMATION

The information presented in Figure 8 and Table 11 provide details about the package branding for the Blackfin processors. For a complete listing of product availability, see Ordering Guide on Page 67.




*Figure 8. Product Information on Package*

**Table 11. Package Brand Information**

| Brand Key | Field Description |
|---|---|
| t | Temperature Range |
| pp | Package Type |
| Z | Lead Free Option (optional) |
| ccc | See Ordering Guide |
| vvvvvv.x | Assembly Lot Code |
| n.n | Silicon Revision |
| yyww | Date Code |

## TIMING SPECIFICATIONS

Table 12 through Table 14 describe the timing requirements for the ADSP-BF534/ADSP-BF536/ADSP-BF537 processor clocks. Take care in selecting MSEL, SSEL, and CSEL ratios so as not to exceed the maximum core clock and system clock. Table 15 describes phase-locked loop operating conditions.

**Table 12. Core Clock Requirements—600 MHz Speed Grade[1]**

| Parameter | | $T_{JUNCTION} \leq 105°C$[2] Min | Max | Unit |
|---|---|---|---|---|
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 1.2 V minimum) | | 600 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 1.045 V minimum) | | 475 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.95 V minimum) | | 425 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.85 V minimum) | | 375 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.8 V) | | 250 | MHz |

[1] The speed grade of a given part is printed on the chip's package as shown in Figure 8 on Page 26 and can also be seen on the specific products ordering guide. It stands for the maximum allowed CCLK frequency at $V_{DDINT}$ = 1.2 V and the maximum allowed VCO frequency at any supply voltage.
[2] See Operating Conditions on Page 24.

**Table 13. Core Clock Requirements—500 MHz Speed Grade[1]**

| Parameter | | $T_{JUNCTION} \leq 105°C$[2] Min | Max | Unit |
|---|---|---|---|---|
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 1.2 V minimum) | | 500 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 1.045 V minimum) | | 444 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.95 V minimum) | | 400 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.85 V minimum) | | 333 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.8 V) | | 250 | MHz |

[1] The speed grade of a given part is printed on the chip's package as shown in Figure 8 on Page 26 and can also be seen on the specific products ordering guide. It stands for the maximum allowed CCLK frequency at $V_{DDINT}$ = 1.2 V and the maximum allowed VCO frequency at any supply voltage.
[2] See Operating Conditions on Page 24.

**Table 14. Core Clock Requirements—400 MHz Speed Grade[1]**

| Parameter | | $120°C \geq T_{JUNCTION} > 105°C$ | | All[2] Other $T_{JUNCTION}$ | | Unit |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 1.14 V minimum) | | 400 | | 400 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 1.045 V minimum) | | 333 | | 363 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.95 V minimum) | | 295 | | 333 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.85 V minimum) | | | | 280 | MHz |
| $f_{CCLK}$ | Core Clock Frequency ($V_{DDINT}$ = 0.8 V) | | | | 250 | MHz |

[1] The speed grade of a given part is printed on the chip's package as shown in Figure 8 on Page 26 and can also be seen on the specific products ordering guide. It stands for the maximum allowed CCLK frequency at $V_{DDINT}$ = 1.2 V and the maximum allowed VCO frequency at any supply voltage.
[2] See Operating Conditions on Page 24.

**Table 15. Phase-Locked Loop Operating Conditions**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| $f_{VCO}$ | Voltage Controlled Oscillator (VCO) Frequency | 50 | Speed Grade[1] | MHz |

[1] The speed grade of a given part is printed on the chip's package as shown in Figure 8 on Page 26 and can also be seen on the specific products ordering guide. It stands for the maximum allowed CCLK frequency at $V_{DDINT}$ = 1.2 V and the maximum allowed VCO frequency at any supply voltage.

Table 16. System Clock Requirements

| Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|
| $f_{SCLK}$ | $V_{DDEXT} = 3.3$ V, $V_{DDINT} \geq 1.14$ V[1] | | 133 | MHz |
| $f_{SCLK}$ | $V_{DDEXT} = 3.3$ V, $V_{DDINT} < 1.14$ V[1] | | 100 | MHz |
| $f_{SCLK}$ | $V_{DDEXT} = 2.5$ V, $V_{DDINT} \geq 1.14$ V[1] | | 133 | MHz |
| $f_{SCLK}$ | $V_{DDEXT} = 2.5$ V, $V_{DDINT} < 1.14$ V[1] | | 100 | MHz |
| $t_{SCLKH}$ | CLKOUT Width High | 2.5 | | ns |
| $t_{SCLKL}$ | CLKOUT Width Low | 2.5 | | ns |

[1] $f_{SCLK}$ is subject to additional restrictions for SDRAM interface operation. See Table 21 on Page 32.

Table 17. Clock Input and Reset Timing

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{CKIN}$ | CLKIN Period[1, 2, 3, 4] | 20.0 | 100.0 | ns |
| $t_{CKINL}$ | CLKIN Low Pulse | 8.0 | | ns |
| $t_{CKINH}$ | CLKIN High Pulse | 8.0 | | ns |
| $t_{BUFDLAY}$ | CLKIN to CLKBUF Delay | | 10 | ns |
| $t_{WRST}$ | $\overline{\text{RESET}}$ Asserted Pulse Width Low[5] | 11 $t_{CKIN}$ | | ns |

[1] Combinations of the CLKIN frequency and the PLL clock multiplier must not exceed the allowed $f_{VCO}$, $f_{CCLK}$, and $f_{SCLK}$ settings discussed in Table 12 through Table 16. Since by default the PLL is multiplying the CLKIN frequency by 10, 300 MHz and 400 MHz speed grade parts can not use the full CLKIN period range.
[2] Applies to PLL bypass mode and PLL nonbypass mode.
[3] CLKIN frequency must not change on the fly.
[4] If the DF bit in the PLL_CTL register is set, then the maximum $t_{CKIN}$ period is 50 ns.
[5] Applies after power-up sequence is complete. At power-up, the processor's internal phase-locked loop requires no more than 2000 CLKIN cycles while $\overline{\text{RESET}}$ is asserted, assuming stable power supplies and CLKIN (not including start-up time of external clock oscillator).

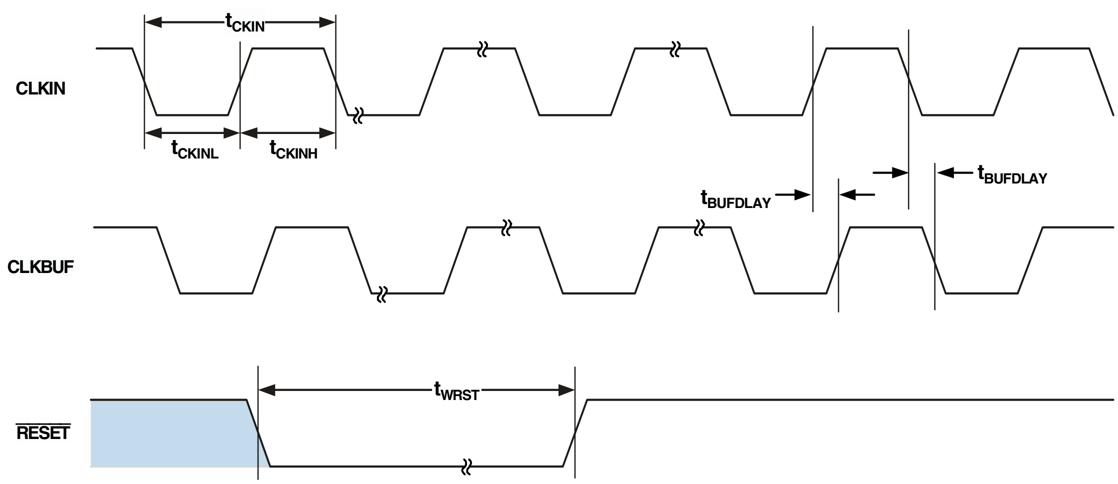



*Figure 9. Clock and Reset Timing*

### *Asynchronous Memory Read Cycle Timing*

**Table 18. Asynchronous Memory Read Cycle Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SDAT}$ | DATA15–0 Setup Before CLKOUT | 2.1 | | ns |
| $t_{HDAT}$ | DATA15–0 Hold After CLKOUT | 0.8 | | ns |
| $t_{SARDY}$ | ARDY Setup Before CLKOUT | 4.0 | | ns |
| $t_{HARDY}$ | ARDY Hold After CLKOUT | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DO}$ | Output Delay After CLKOUT[1] | | 6.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT [1] | 0.8 | | ns |

[1] Output pins include $\overline{AMS3-0}$, $\overline{ABE1-0}$, ADDR19–1, $\overline{AOE}$, $\overline{ARE}$.

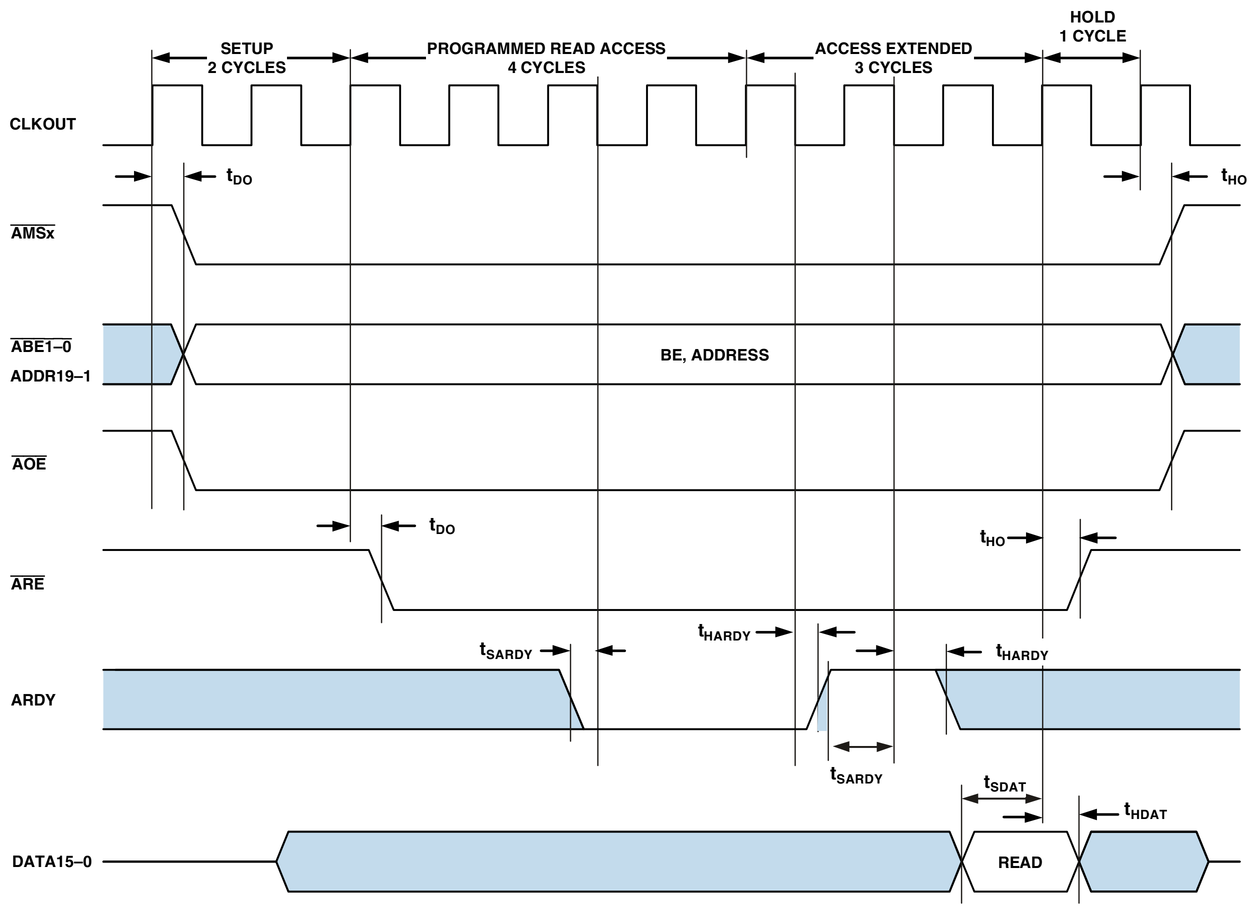



*Figure 10. Asynchronous Memory Read Cycle Timing*

### *Asynchronous Memory Write Cycle Timing*

**Table 19. Asynchronous Memory Write Cycle Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SARDY}$ | ARDY Setup Before CLKOUT | 4.0 | | ns |
| $t_{HARDY}$ | ARDY Hold After CLKOUT | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DDAT}$ | DATA15–0 Disable After CLKOUT | | 6.0 | ns |
| $t_{ENDAT}$ | DATA15–0 Enable After CLKOUT | 1.0 | | ns |
| $t_{DO}$ | Output Delay After CLKOUT[1] | | 6.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT [1] | 0.8 | | ns |

[1] Output pins include $\overline{\text{AMS3–0}}$, $\overline{\text{ABE1–0}}$, ADDR19–1, DATA15–0, $\overline{\text{AOE}}$, $\overline{\text{AWE}}$.

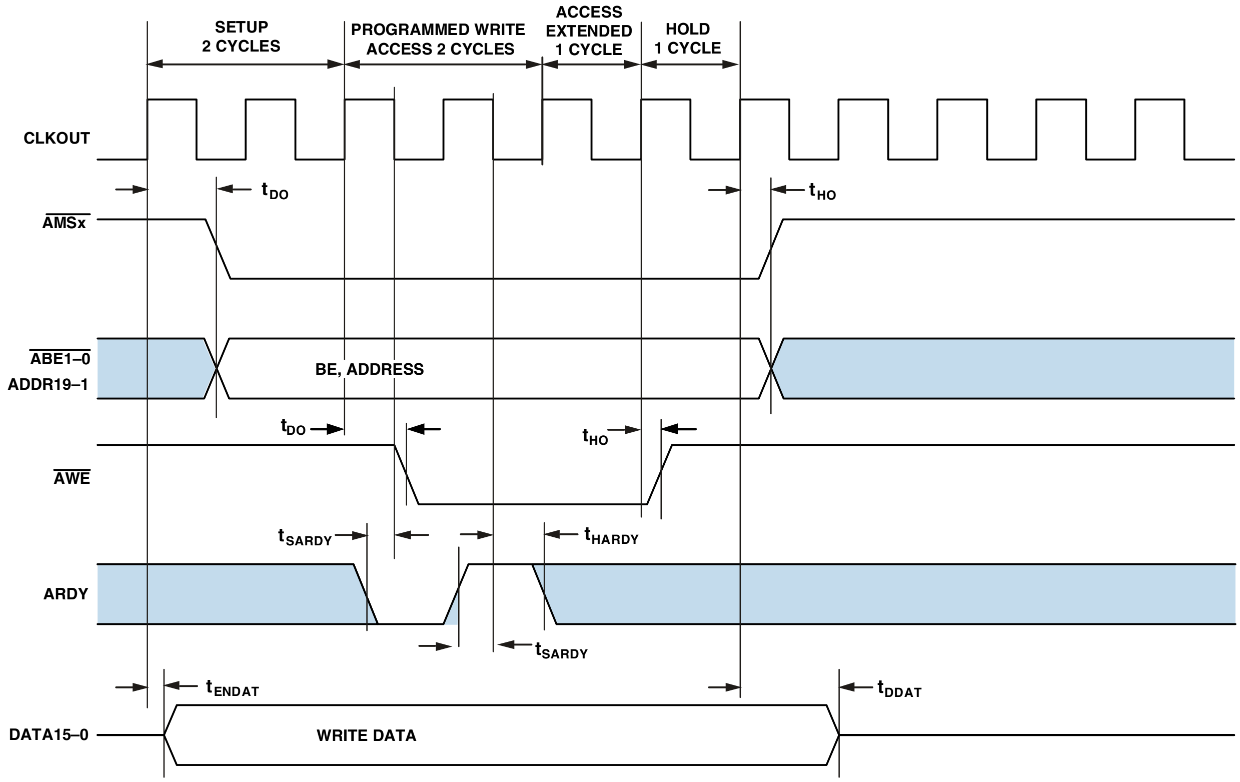



*Figure 11. Asynchronous Memory Write Cycle Timing*

### External Port Bus Request and Grant Cycle Timing

Table 20 and Figure 12 describe external port bus request and bus grant operations.

**Table 20. External Port Bus Request and Grant Cycle Timing**

| Parameter[1, 2] | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{BS}$ | $\overline{BR}$ Asserted to CLKOUT Low Setup | 4.6 | | ns |
| $t_{BH}$ | CLKOUT Low to $\overline{BR}$ Deasserted Hold Time | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{SD}$ | CLKOUT Low to $\overline{AMSx}$, Address, and $\overline{RD}/\overline{WR}$ Disable | | 4.5 | ns |
| $t_{SE}$ | CLKOUT Low to $\overline{AMSx}$, Address, and $\overline{RD}/\overline{WR}$ Enable | | 4.5 | ns |
| $t_{DBG}$ | CLKOUT High to $\overline{BG}$ Asserted Setup | | 3.6 | ns |
| $t_{EBG}$ | CLKOUT High to $\overline{BG}$ Deasserted Hold Time | | 3.6 | ns |
| $t_{DBH}$ | CLKOUT High to $\overline{BGH}$ Asserted Setup | | 3.6 | ns |
| $t_{EBH}$ | CLKOUT High to $\overline{BGH}$ Deasserted Hold Time | | 3.6 | ns |

[1] These are preliminary timing parameters that are based on worst-case operating conditions.
[2] The pad loads for these timing parameters are 20 pF.

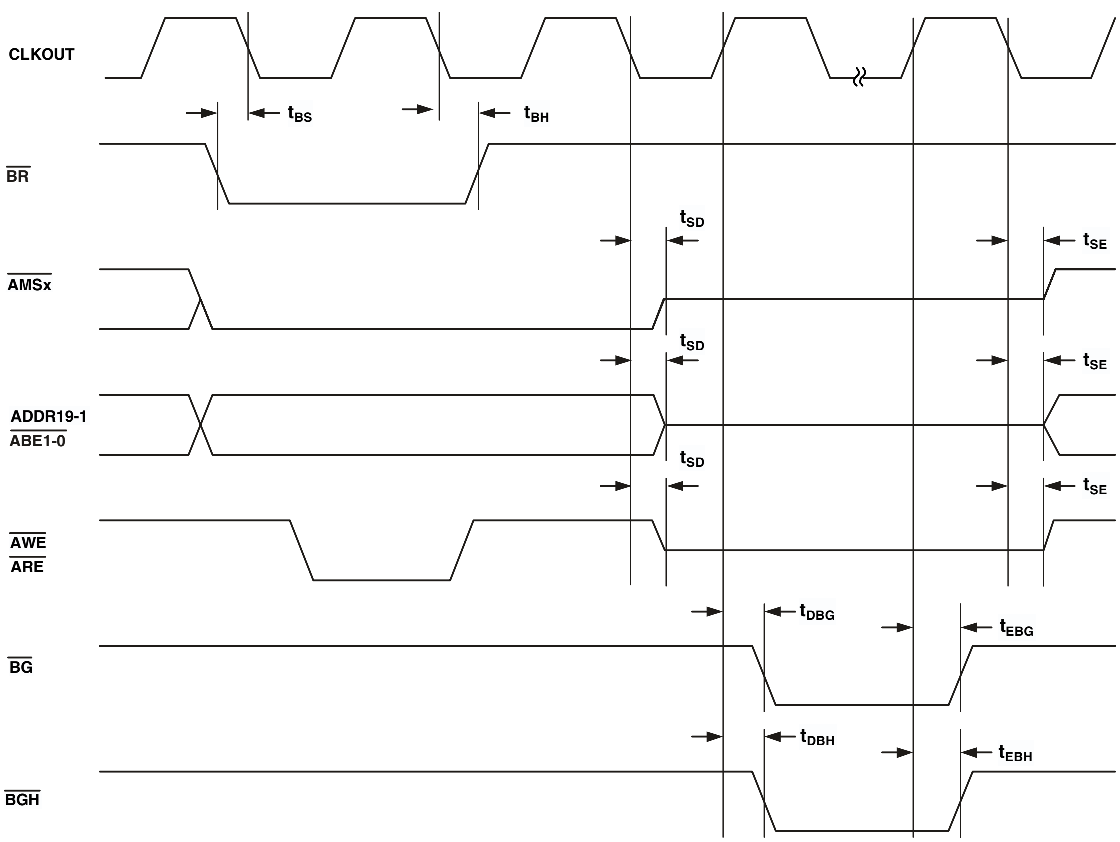



*Figure 12. External Port Bus Request and Grant Cycle Timing*

### *SDRAM Interface Timing*

**Table 21. SDRAM Interface Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SSDAT}$ | $DATA_{15–0}$ Setup Before CLKOUT | 1.5 | | ns |
| $t_{HSDAT}$ | $DATA_{15–0}$ Hold After CLKOUT | 0.8 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DCAD}$ | COMMAND, $ADDR_{19–1}$, $DATA_{15–0}$ Delay After CLKOUT[1] | | 4.0 | ns |
| $t_{HCAD}$ | COMMAND, $ADDR_{19–1}$, $DATA_{15–0}$ Hold After CLKOUT[1] | 1.0 | | ns |
| $t_{DSDAT}$ | $DATA_{15–0}$ Disable After CLKOUT | | 6.0 | ns |
| $t_{ENSDAT}$ | $DATA_{15–0}$ Enable After CLKOUT | 1.0 | | ns |
| $t_{SCLK}$ | CLKOUT Period $T_J \leq +105°C$[2] | 7.5 | | ns |
| $t_{SCLK}$ | CLKOUT Period $T_J > +105°C$[2] | 10 | | ns |

[1] Command pins include: $\overline{SRAS}$, $\overline{SCAS}$, $\overline{SWE}$, SDQM, $\overline{SMS}$, SA10, SCKE.
[2] These limits are specific to the SDRAM interface only. In addition, CLKOUT must always comply with the limits in Table 16 on Page 28.

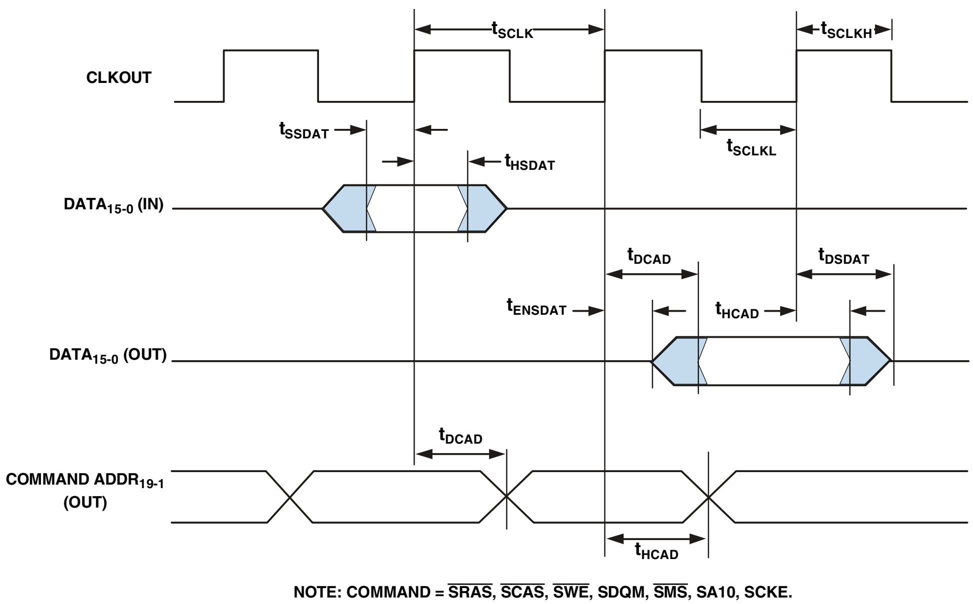



*Figure 13. SDRAM Interface Timing*

### *External DMA Request Timing*

Table 22 and Figure 14 describe the external DMA request operations.

**Table 22. External DMA Request Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{DR}$ | DMARx Asserted to CLKOUT High Setup | 6.0 | | ns |
| $t_{DH}$ | CLKOUT High to DMARx Deasserted Hold Time | 0.0 | | ns |
| $t_{DMARACT}$ | DMARx Active Pulse Width | $1.0 \times t_{SCLK}$ | | ns |
| $t_{DMARINACT}$ | DMARx Inactive Pulse Width | $1.75 \times t_{SCLK}$ | | ns |

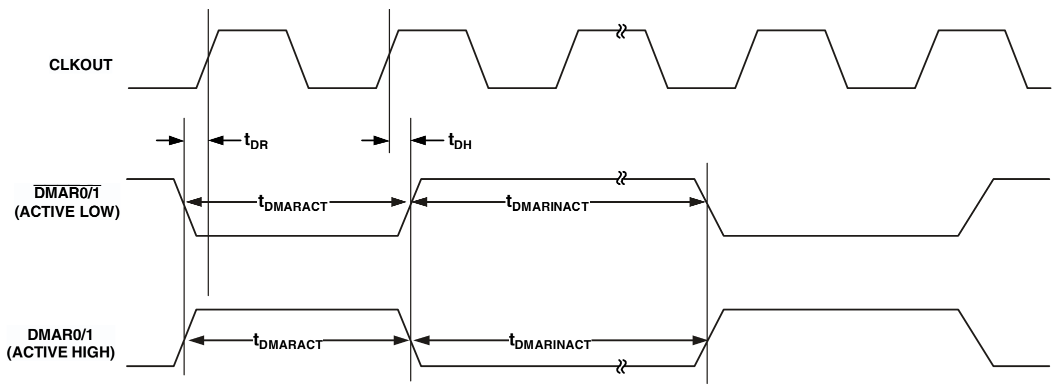



*Figure 14. External DMA Request Timing*

### *Parallel Peripheral Interface Timing*

Table 23 and Figure 15 on Page 34, Figure 19 on Page 38, and Figure 20 on Page 39 describe parallel peripheral interface operations.

**Table 23. Parallel Peripheral Interface Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{PCLKW}$ | PPI_CLK Width[1] | 6.0 | | ns |
| $t_{PCLK}$ | PPI_CLK Period[1] | 15.0 | | ns |
| *Timing Requirements—GP Input and Frame Capture Modes* | | | | |
| $t_{SFSPE}$ | External Frame Sync Setup Before PPI_CLK (Nonsampling Edge for Rx, Sampling Edge for Tx) | 6.7 | | ns |
| $t_{HFSPE}$ | External Frame Sync Hold After PPI_CLK | 1.0 | | ns |
| $t_{SDRPE}$ | Receive Data Setup Before PPI_CLK | 3.5 | | ns |
| $t_{HDRPE}$ | Receive Data Hold After PPI_CLK | 1.5 | | ns |
| *Switching Characteristics—GP Output and Frame Capture Modes* | | | | |
| $t_{DFSPE}$ | Internal Frame Sync Delay After PPI_CLK | | 8.0 | ns |
| $t_{HOFSPE}$ | Internal Frame Sync Hold After PPI_CLK | 1.7 | | ns |
| $t_{DDTPE}$ | Transmit Data Delay After PPI_CLK | | 8.0 | ns |
| $t_{HDTPE}$ | Transmit Data Hold After PPI_CLK | 1.8 | | ns |

[1] PPI_CLK frequency cannot exceed $f_{SCLK}/2$.

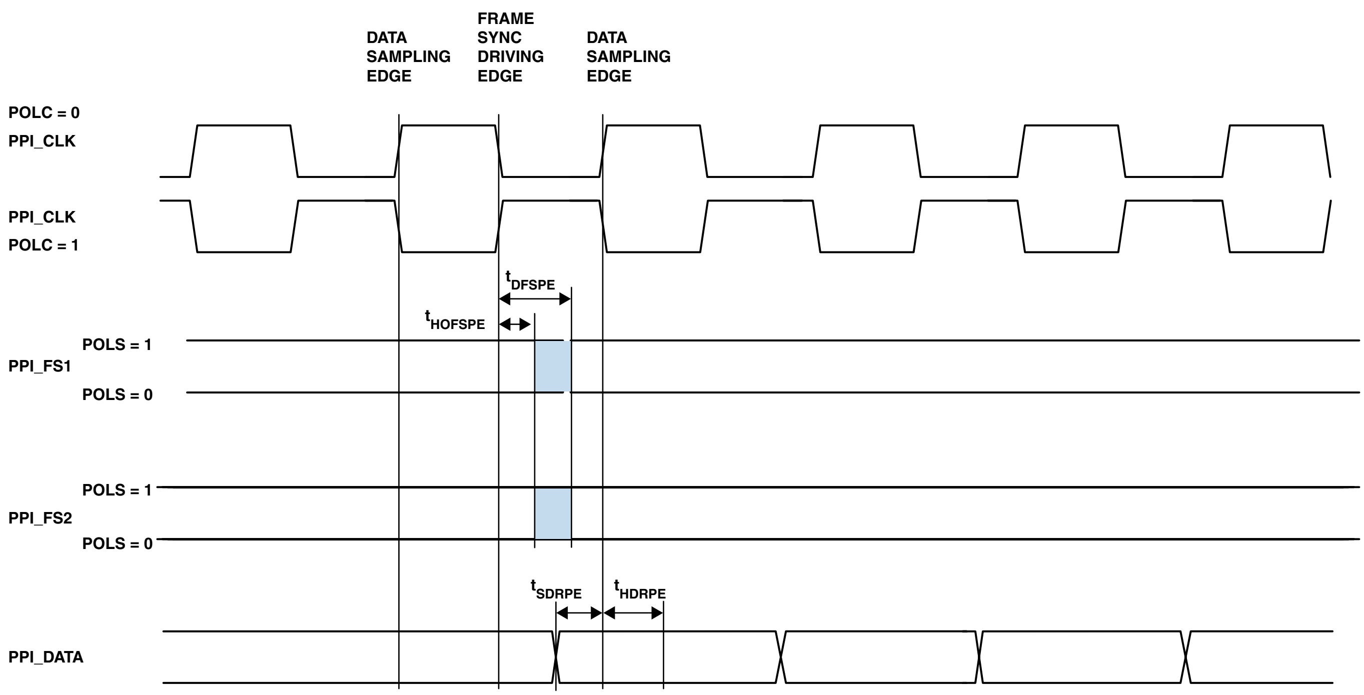



*Figure 15. PPI GP Rx Mode with Internal Frame Sync Timing*

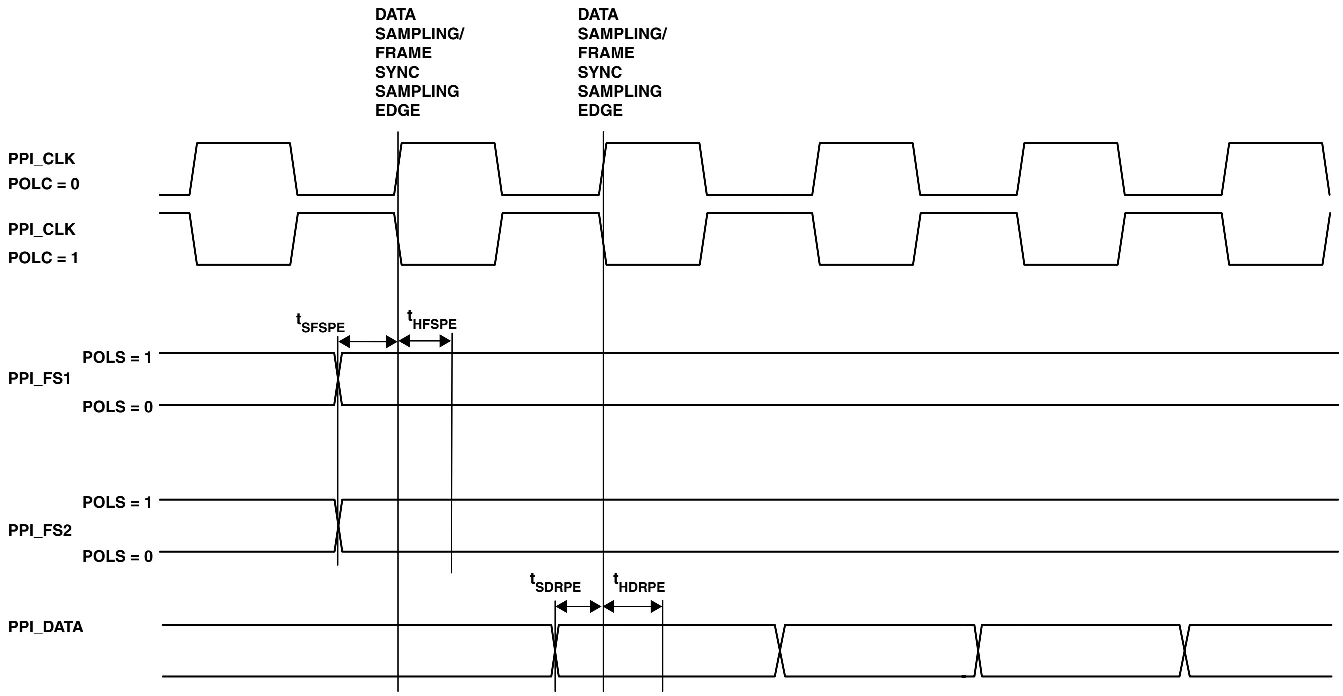



*Figure 16. PPI GP Rx Mode with External Frame Sync Timing*

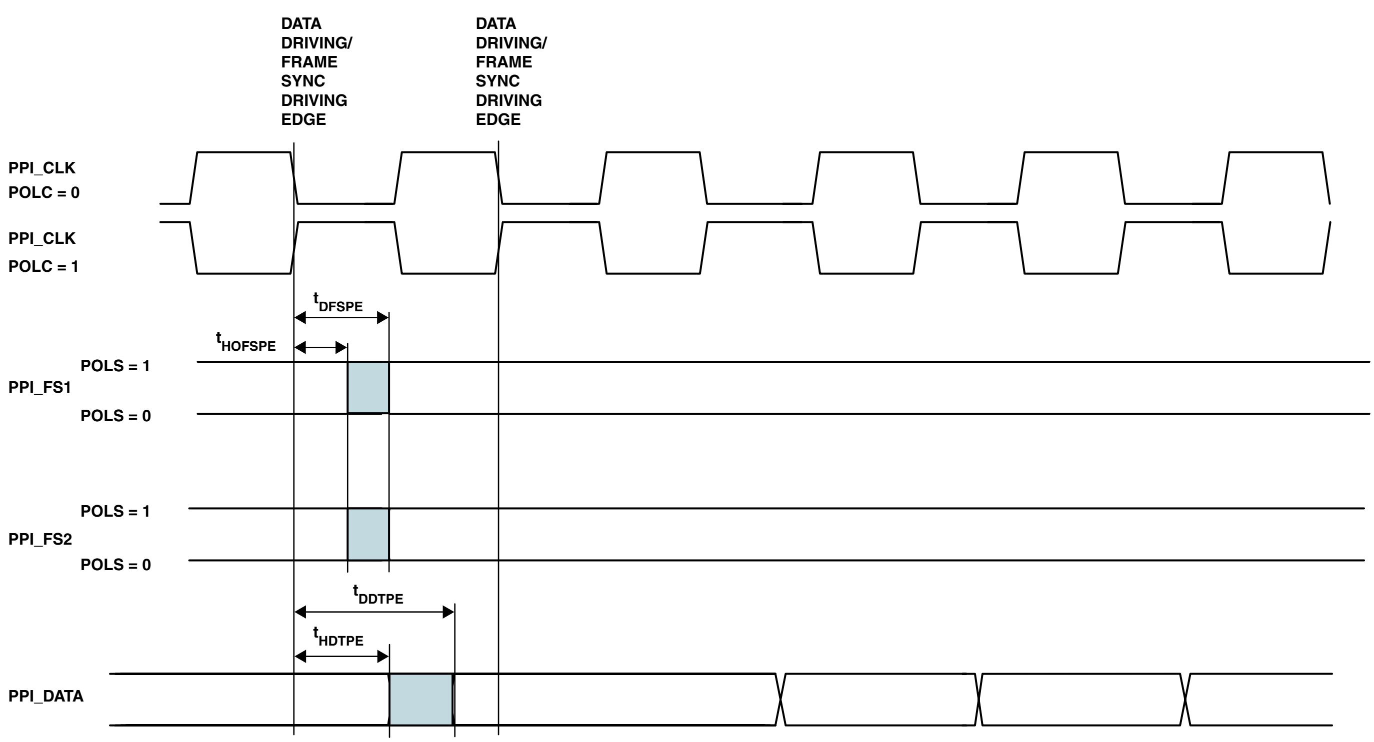



*Figure 17. PPI GP Tx Mode with Internal Frame Sync Timing*

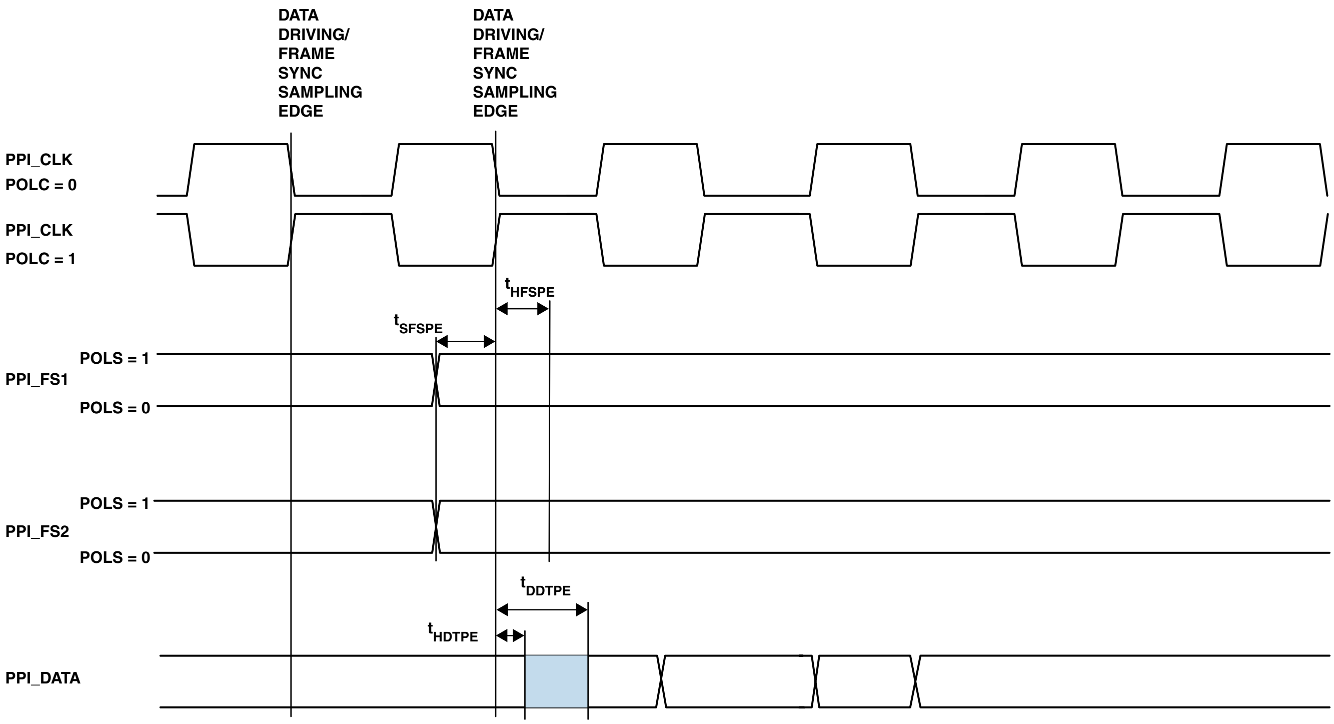



*Figure 18. PPI GP Tx Mode with External Frame Sync Timing*

### Serial Ports

Table 24 through Table 27 on Page 38 and Figure 19 on Page 38 through Figure 21 on Page 40 describe serial port operations.

**Table 24. Serial Ports—External Clock**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SFSE}$ | TFS/RFS Setup Before TSCLK/RSCLK[1] | 3.0 | | ns |
| $t_{HFSE}$ | TFS/RFS Hold After TSCLK/RSCLK[1] | 3.0 | | ns |
| $t_{SDRE}$ | Receive Data Setup Before RSCLK[1] | 3.0 | | ns |
| $t_{HDRE}$ | Receive Data Hold After RSCLK[1] | 3.0 | | ns |
| $t_{SCLKEW}$ | TSCLK/RSCLK Width | 4.5 | | ns |
| $t_{SCLKE}$ | TSCLK/RSCLK Period | 15.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DFSE}$ | TFS/RFS Delay After TSCLK/RSCLK (Internally Generated TFS/RFS)[2] | | 10.0 | ns |
| $t_{HOFSE}$ | TFS/RFS Hold After TSCLK/RSCLK (Internally Generated TFS/RFS)[2] | 0 | | ns |
| $t_{DDTE}$ | Transmit Data Delay After TSCLK[2] | | 10.0 | ns |
| $t_{HDTE}$ | Transmit Data Hold After TSCLK[2] | 0 | | ns |

[1] Referenced to sample edge.
[2] Referenced to drive edge.

**Table 25. Serial Ports—Internal Clock**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SFSI}$ | TFS/RFS Setup Before TSCLK/RSCLK[1] | 8.0 | | ns |
| $t_{HFSI}$ | TFS/RFS Hold After TSCLK/RSCLK[1] | –1.5 | | ns |
| $t_{SDRI}$ | Receive Data Setup Before RSCLK[1] | 8.0 | | ns |
| $t_{HDRI}$ | Receive Data Hold After RSCLK[1] | –1.5 | | ns |
| $t_{SCLKEW}$ | TSCLK/RSCLK Width | 4.5 | | ns |
| $t_{SCLKE}$ | TSCLK/RSCLK Period | 15.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DFSI}$ | TFS/RFS Delay After TSCLK/RSCLK (Internally Generated TFS/RFS)[2] | | 3.0 | ns |
| $t_{HOFSI}$ | TFS/RFS Hold After TSCLK/RSCLK (Internally Generated TFS/RFS)[2] | –1.0 | | ns |
| $t_{DDTI}$ | Transmit Data Delay After TSCLK[2] | | 3.0 | ns |
| $t_{HDTI}$ | Transmit Data Hold After TSCLK[2] | –1.0 | | ns |
| $t_{SCLKIW}$ | TSCLK/RSCLK Width | 4.5 | | ns |

[1] Referenced to sample edge.
[2] Referenced to drive edge.

**Table 26. Serial Ports—Enable and Three-State**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DTENE}$ | Data Enable Delay from External TSCLK[1] | 0 | | ns |
| $t_{DDTTE}$ | Data Disable Delay from External TSCLK[1] | | 10.0 | ns |
| $t_{DTENI}$ | Data Enable Delay from Internal TSCLK[1] | –2.0 | | ns |
| $t_{DDTTI}$ | Data Disable Delay from Internal TSCLK[1] | | 3.0 | ns |

[1] Referenced to drive edge.

**Table 27. External Late Frame Sync**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DDTLFSE}$ | Data Delay from Late External TFS or External RFS with MCE = 1, MFD = 0[1,2] | | 10.0 | ns |
| $t_{DTENLFS}$ | Data Enable from Late FS or MCE = 1, MFD = 0[1,2] | 0 | | ns |

[1] MCE = 1, TFS enable and TFS valid follow $t_{DDTENFS}$ and $t_{DDTLFS}$.
[2] If external RFS/TFS setup to RSCLK/TSCLK > $t_{SCLKE}/2$, then $t_{DDTE/I}$ and $t_{DTENE/I}$ apply, otherwise $t_{DDTLFSE}$ and $t_{DTENLFS}$ apply.

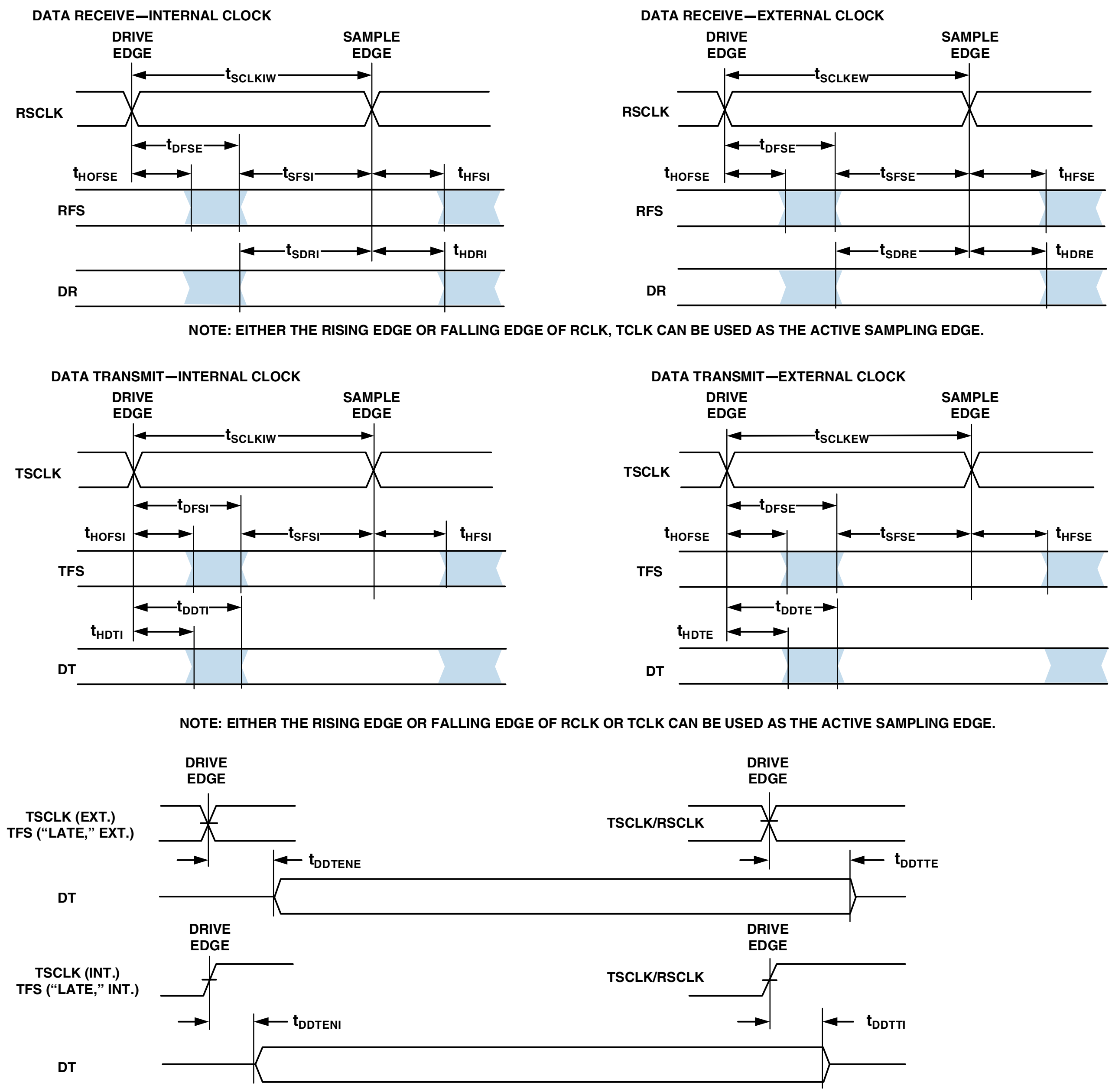



*Figure 19. Serial Ports*

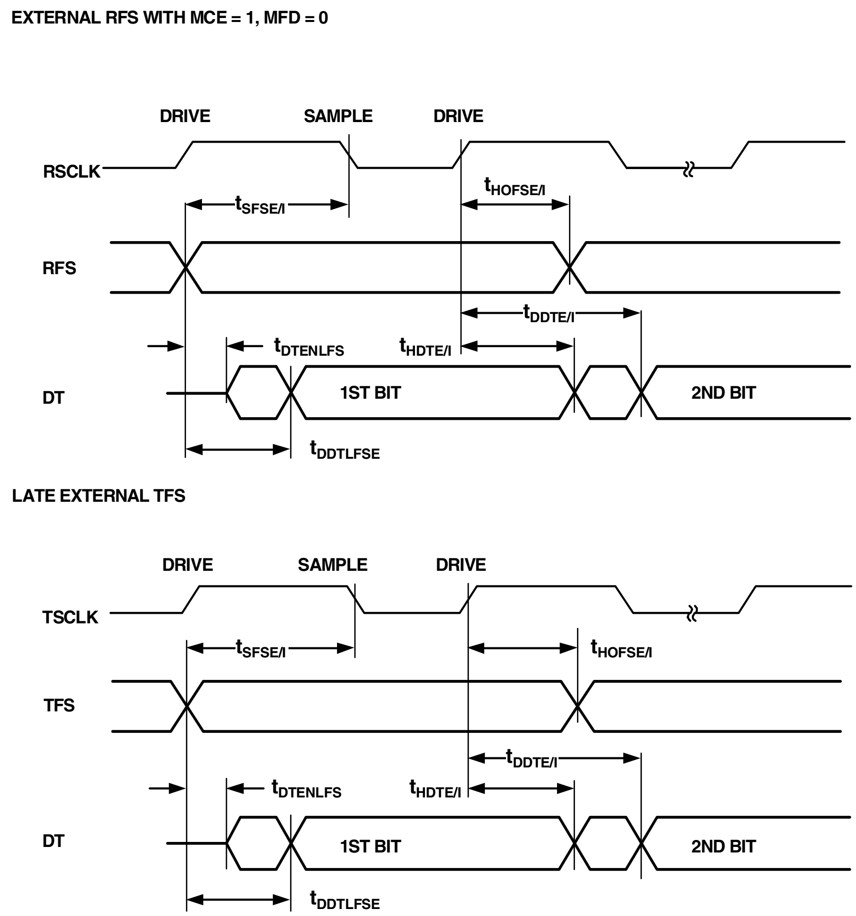



*Figure 20. External Late Frame Sync (Frame Sync Setup < $t_{SCLKE}$/2)*

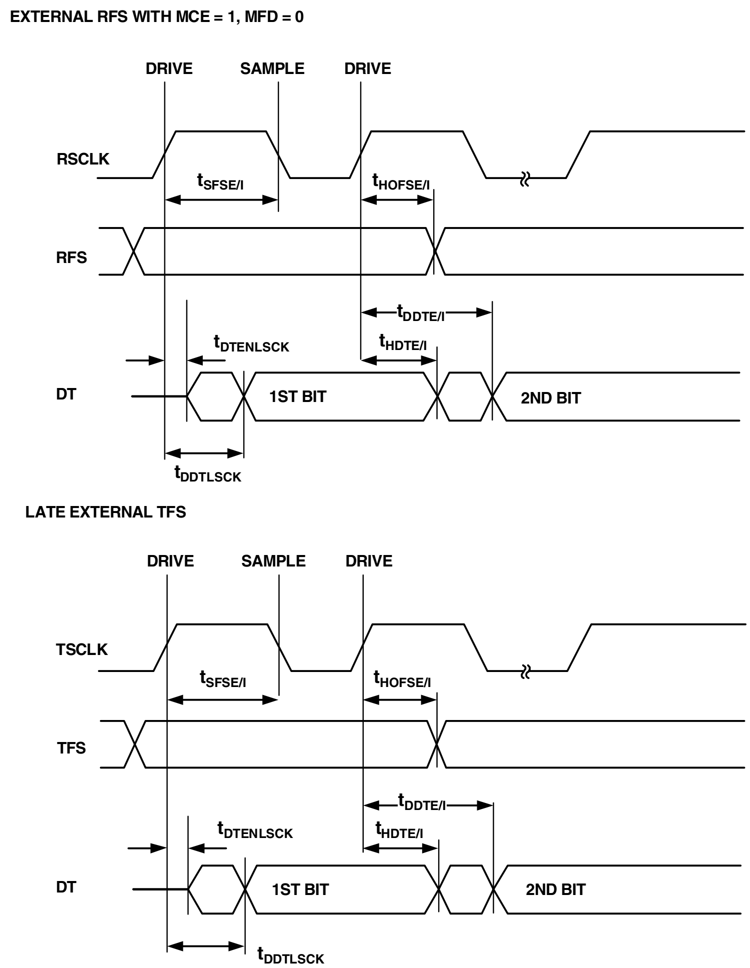



*Figure 21. External Late Frame Sync (Frame Sync Setup > $t_{SCLKE}/2$)*

### *Serial Peripheral Interface Port—Master Timing*

Table 28 and Figure 22 describe SPI port master operations.

**Table 28. Serial Peripheral Interface (SPI) Port—Master Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SSPIDM}$ | Data Input Valid to SCK Edge (Data Input Setup) | 7.5 | | ns |
| $t_{HSPIDM}$ | SCK Sampling Edge to Data Input Invalid | –1.5 | | ns |
| *Switching Characteristics* | | | | |
| $t_{SDSCIM}$ | $\overline{\text{SPISELx}}$ Low to First SCK Edge (x = 0 or x = 1) | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SPICHM}$ | Serial Clock High Period | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SPICLM}$ | Serial Clock Low Period | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SPICLK}$ | Serial Clock Period | $4 \times t_{SCLK} - 1.5$ | | ns |
| $t_{HDSM}$ | Last SCK Edge to $\overline{\text{SPISELx}}$ High (x = 0 or x = 1) | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SPITDM}$ | Sequential Transfer Delay | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{DDSPIDM}$ | SCK Edge to Data Out Valid (Data Out Delay) | 0 | 6 | ns |
| $t_{HDSPIDM}$ | SCK Edge to Data Out Invalid (Data Out Hold) | –1.0 | +4.0 | ns |

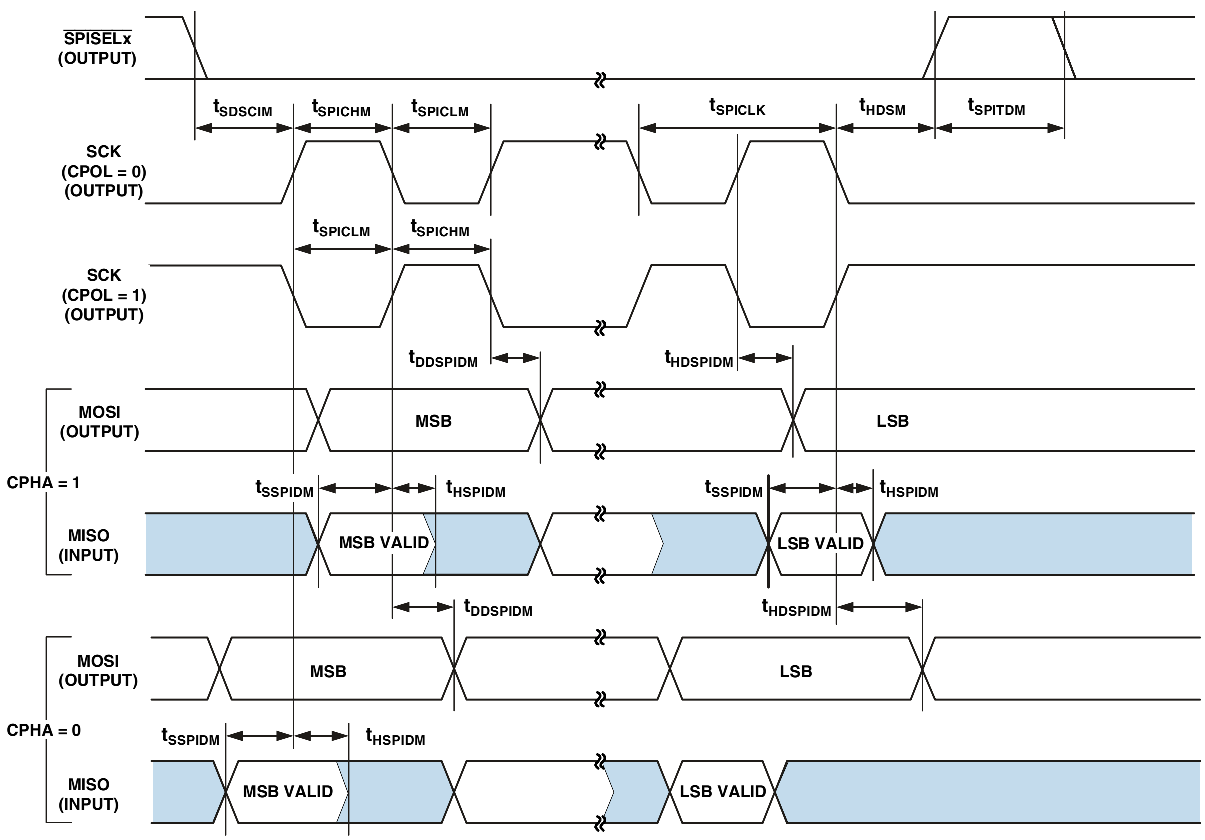



*Figure 22. Serial Peripheral Interface (SPI) Port—Master Timing*

### *Serial Peripheral Interface Port—Slave Timing*

Table 29 and Figure 23 describe SPI port slave operations.

**Table 29. Serial Peripheral Interface (SPI) Port—Slave Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SPICHS}$ | Serial Clock High Period | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SPICLS}$ | Serial Clock Low Period | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SPICLK}$ | Serial Clock Period | $4 \times t_{SCLK} - 1.5$ | | ns |
| $t_{HDS}$ | Last SCK Edge to $\overline{SPISS}$ Not Asserted | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SPITDS}$ | Sequential Transfer Delay | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SDSCI}$ | $\overline{SPISS}$ Assertion to First SCK Edge | $2 \times t_{SCLK} - 1.5$ | | ns |
| $t_{SSPID}$ | Data Input Valid to SCK Edge (Data Input Setup) | 1.6 | | ns |
| $t_{HSPID}$ | SCK Sampling Edge to Data Input Invalid | 1.6 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DSOE}$ | $\overline{SPISS}$ Assertion to Data Out Active | 0 | 8 | ns |
| $t_{DSDHI}$ | $\overline{SPISS}$ Deassertion to Data High Impedance | 0 | 8 | ns |
| $t_{DDSPID}$ | SCK Edge to Data Out Valid (Data Out Delay) | 0 | 10 | ns |
| $t_{HDSPID}$ | SCK Edge to Data Out Invalid (Data Out Hold) | 0 | 10 | ns |

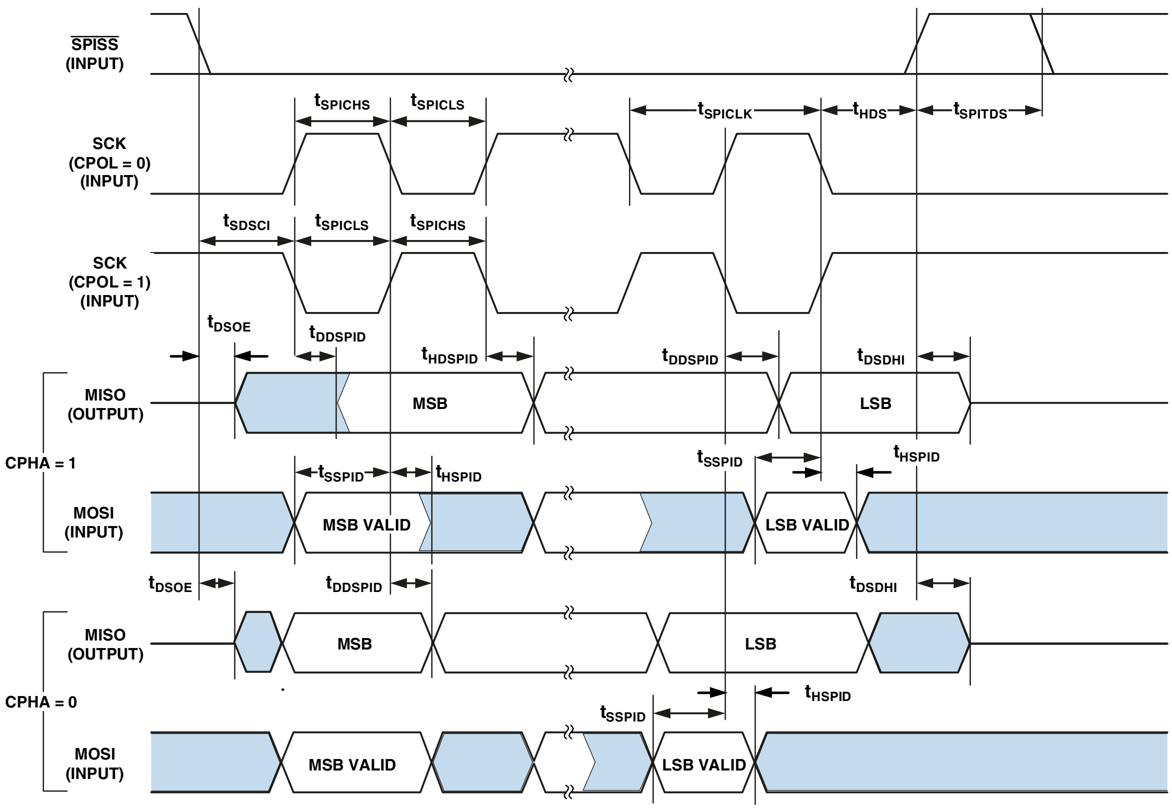



*Figure 23. Serial Peripheral Interface (SPI) Port—Slave Timing*

### *Universal Asynchronous Receiver-Transmitter (UART) Ports—Receive and Transmit Timing*

Figure 24 describes the UART ports receive and transmit operations. The maximum baud rate is SCLK/16. As shown in Figure 24 there is some latency between the generation of internal UART interrupts and the external data operations. These latencies are negligible at the data transmission rates for the UART.

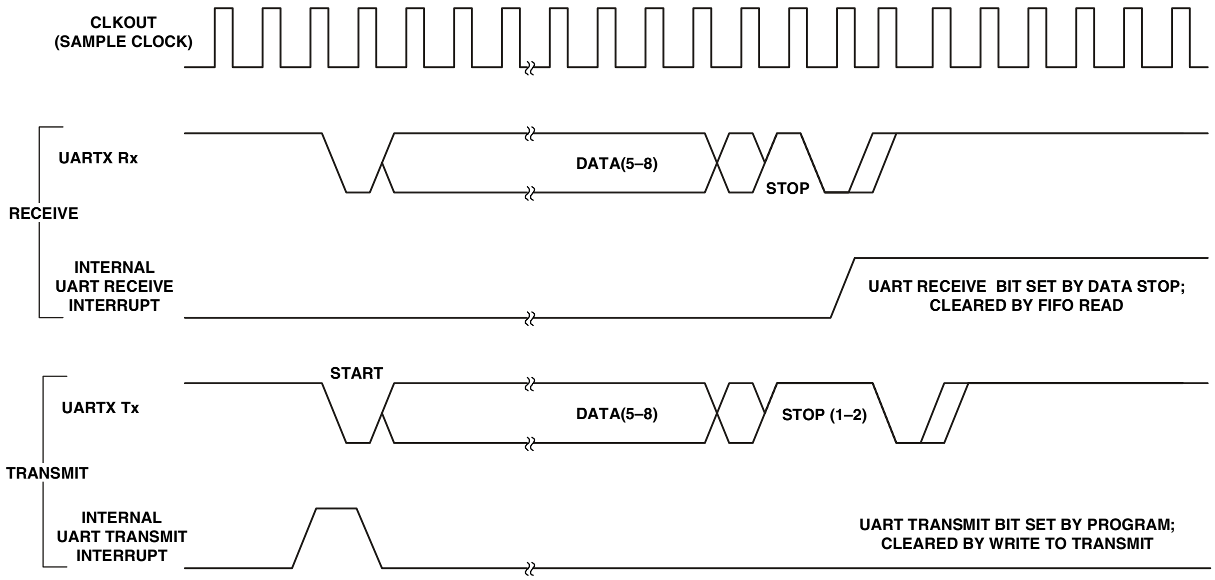



*Figure 24. UART Ports—Receive and Transmit Timing*

### General-Purpose Port Timing

Table 30 and Figure 25 describe general-purpose port operations.

**Table 30. General-Purpose Port Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirement* | | | | |
| $t_{WFI}$ | General-Purpose Port Pin Input Pulse Width | $t_{SCLK}$ + 1 | | ns |
| *Switching Characteristic* | | | | |
| $t_{GPOD}$ | General-Purpose Port Pin Output Delay from CLKOUT Low | 0 | 6 | ns |

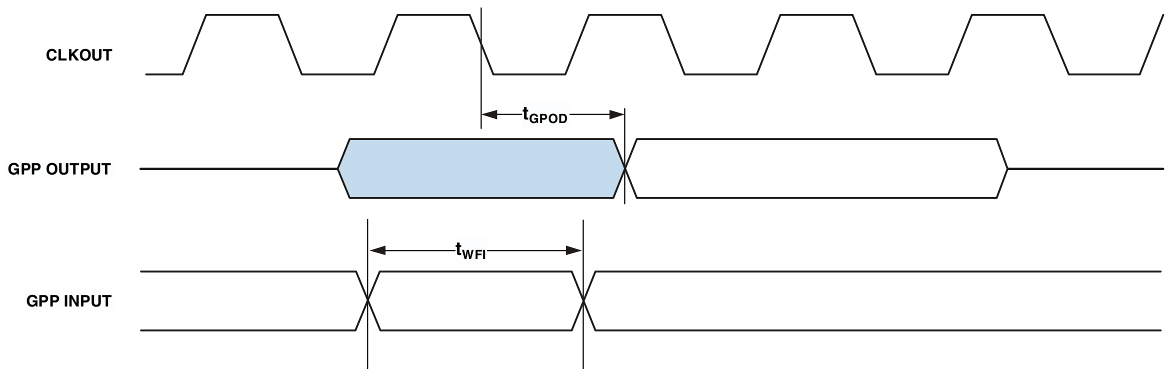



*Figure 25. General-Purpose Port Timing*

### *Timer Cycle Timing*

Table 31 and Figure 26 describe timer expired operations. The input signal is asynchronous in "width capture mode" and "external clock mode" and has an absolute maximum input frequency of ($f_{SCLK}$/2) MHz.

**Table 31. Timer Cycle Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Characteristics* | | | | |
| $t_{WL}$ | Timer Pulse Width Input Low (Measured In SCLK Cycles)[1] | $1 \times t_{SCLK}$ | | ns |
| $t_{WH}$ | Timer Pulse Width Input High (Measured In SCLK Cycles)[1] | $1 \times t_{SCLK}$ | | ns |
| $t_{TIS}$ | Timer Input Setup Time Before CLKOUT Low[2] | 5 | | ns |
| $t_{TIH}$ | Timer Input Hold Time After CLKOUT Low[2] | –2 | | ns |
| *Switching Characteristics* | | | | |
| $t_{HTO}$ | Timer Pulse Width Output (Measured In SCLK Cycles) | $1 \times t_{SCLK}$ | $(2^{32}–1) \times t_{SCLK}$ | ns |
| $t_{TOD}$ | Timer Output Update Delay After CLKOUT High | | 6 | ns |

[1] The minimum pulse widths apply for TMRx signals in width capture and external clock modes. They also apply to the PF15 or PPI_CLK signals in PWM output mode.
[2] Either a valid setup and hold time or a valid pulse width is sufficient. There is no need to resynchronize programmable flag inputs.

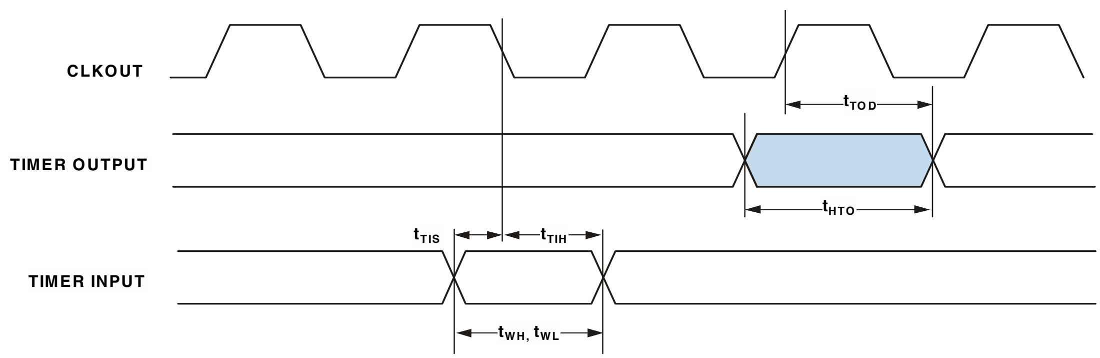



*Figure 26. Timer Cycle Timing*

### *Timer Clock Timing*

Table 32 and Figure 27 describe timer clock timing.

**Table 32. Timer Clock Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristic* | | | | |
| $t_{TODP}$ | Timer Output Update Delay After PPICLK High | | 12 | ns |

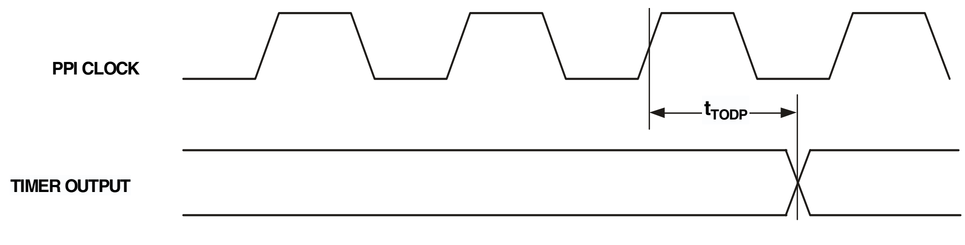



*Figure 27. Timer Clock Timing*

### *JTAG Test and Emulation Port Timing*

Table 33 and Figure 28 describe JTAG port operations.

**Table 33. JTAG Port Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Parameters* | | | | |
| $t_{TCK}$ | TCK Period | 20 | | ns |
| $t_{STAP}$ | TDI, TMS Setup Before TCK High | 4 | | ns |
| $t_{HTAP}$ | TDI, TMS Hold After TCK High | 4 | | ns |
| $t_{SSYS}$ | System Inputs Setup Before TCK High[1] | 4 | | ns |
| $t_{HSYS}$ | System Inputs Hold After TCK High[1] | 5 | | ns |
| $t_{TRSTW}$ | $\overline{\text{TRST}}$ Pulse Width[2] (Measured in TCK Cycles) | 4 | | TCK |
| *Switching Characteristics* | | | | |
| $t_{DTDO}$ | TDO Delay From TCK Low | | 10 | ns |
| $t_{DSYS}$ | System Outputs Delay After TCK Low[3] | 0 | 12 | ns |

[1] System Inputs = DATA15–0, $\overline{\text{BR}}$, ARDY, SCL, SDA, TFS0, TSCLK0, RSCLK0, RFS0, DR0PRI, DR0SEC, PF15–0, PG15–0, PH15–0, MDIO, TCK, TD1, TMS, $\overline{\text{TRST}}$, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, BMODE2–0.

[2] 50 MHz maximum

[3] System Outputs = DATA15–0, ADDR19–1, $\overline{\text{ABE1–0}}$, $\overline{\text{AOE}}$, $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, $\overline{\text{AMS3–0}}$, $\overline{\text{SRAS}}$, $\overline{\text{SCAS}}$, $\overline{\text{SWE}}$, SCKE, CLKOUT, SA10, $\overline{\text{SMS}}$, SCL, SDA, TSCLK0, TFS0, RFS0, RSCLK0, DT0PRI, DT0SEC, PF15–0, PG15–0, PH15–0, RTX0, TD0, $\overline{\text{EMU}}$, XTAL, VROUT.

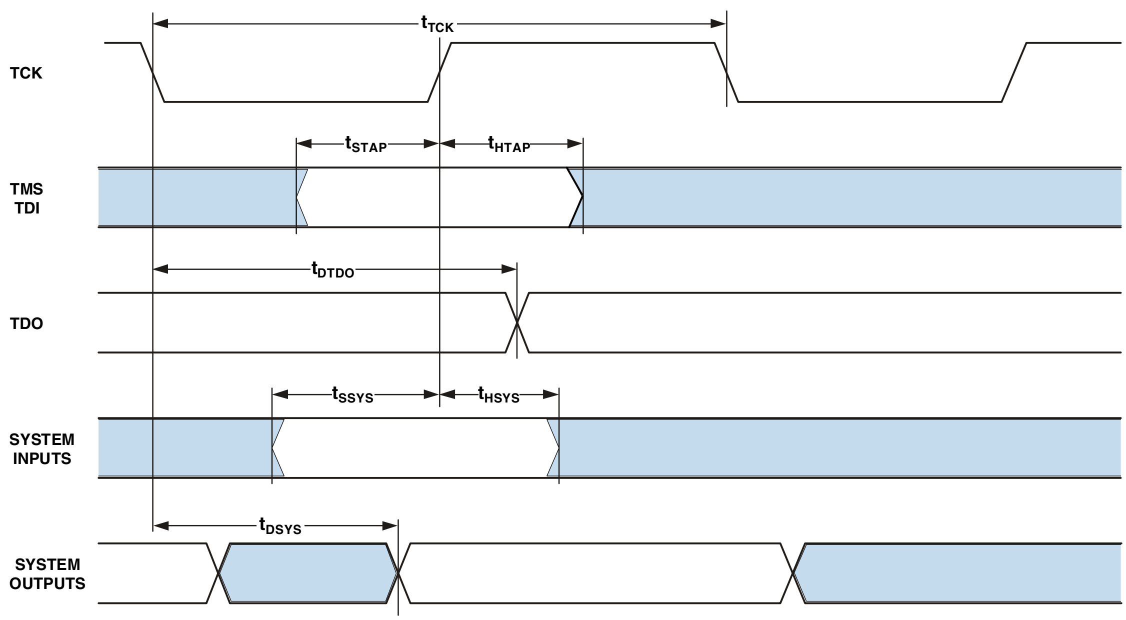



*Figure 28. JTAG Port Timing*

### 10/100 Ethernet MAC Controller Timing

Table 34 through Table 39 and Figure 29 through Figure 34 describe the 10/100 Ethernet MAC controller operations. This feature is only available on the ADSP-BF536 and ADSP-BF537 processors. For more information, see Table 1 on Page 3.

**Table 34. 10/100 Ethernet MAC Controller Timing: MII Receive Signal**

| Parameter[1] | | Min | Max | Unit |
|---|---|---|---|---|
| $t_{ERXCLKF}$ | ERxCLK Frequency ($f_{SCLK}$ = SCLK Frequency) | None | 25 MHz + 1%<br>$f_{SCLK}$ + 1% | ns |
| $t_{ERXCLKW}$ | ERxCLK Width ($t_{ERxCLK}$ = ERxCLK Period) | $t_{ERxCLK} \times 35\%$ | $t_{ERxCLK} \times 65\%$ | ns |
| $t_{ERXCLKIS}$ | Rx Input Valid to ERxCLK Rising Edge (Data In Setup) | 7.5 | | ns |
| $t_{ERXCLKIH}$ | ERxCLK Rising Edge to Rx Input Invalid (Data In Hold) | 7.5 | | ns |

[1] MII inputs synchronous to ERxCLK are ERxD3–0, ERxDV, and ERxER.

**Table 35. 10/100 Ethernet MAC Controller Timing: MII Transmit Signal**

| Parameter[1] | | Min | Max | Unit |
|---|---|---|---|---|
| $t_{ETF}$ | ETxCLK Frequency ($f_{SCLK}$ = SCLK Frequency) | None | 25 MHz + 1%<br>$f_{SCLK}$ + 1% | ns |
| $t_{ETXCLKW}$ | ETxCLK Width ($t_{ETxCLK}$ = ETxCLK Period) | $t_{ETxCLK} \times 35\%$ | $t_{ETxCLK} \times 65\%$ | ns |
| $t_{ETXCLKOV}$ | ETxCLK Rising Edge to Tx Output Valid (Data Out Valid) | | 20 | ns |
| $t_{ETXCLKOH}$ | ETxCLK Rising Edge to Tx Output Invalid (Data Out Hold) | 0 | | ns |

[1] MII outputs synchronous to ETxCLK are ETxD3–0.

**Table 36. 10/100 Ethernet MAC Controller Timing: RMII Receive Signal**

| Parameter[1] | | Min | Max | Unit |
|---|---|---|---|---|
| $t_{EREFCLKF}$ | REF_CLK Frequency ($f_{SCLK}$ = SCLK Frequency) | None | 50 MHz + 1%<br>$2 \times f_{SCLK}$ + 1% | ns |
| $t_{EREFCLKW}$ | EREF_CLK Width ($t_{EREFCLK}$ = EREFCLK Period) | $t_{EREFCLK} \times 35\%$ | $t_{EREFCLK} \times 65\%$ | ns |
| $t_{EREFCLKIS}$ | Rx Input Valid to RMII REF_CLK Rising Edge (Data In Setup) | 4 | | ns |
| $t_{EREFCLKIH}$ | RMII REF_CLK Rising Edge to Rx Input Invalid (Data In Hold) | 2 | | ns |

[1] RMII inputs synchronous to RMII REF_CLK are ERxD1–0, RMII CRS_DV, and ERxER.

**Table 37. 10/100 Ethernet MAC Controller Timing: RMII Transmit Signal**

| Parameter[1] | | Min | Max | Unit |
|---|---|---|---|---|
| $t_{EREFCLKOV}$ | RMII REF_CLK Rising Edge to Tx Output Valid (Data Out Valid) | | 7.5 | ns |
| $t_{EREFCLKOH}$ | RMII REF_CLK Rising Edge to Tx Output Invalid (Data Out Hold) | 2 | | ns |

[1] RMII outputs synchronous to RMII REF_CLK are ETxD1–0.

**Table 38. 10/100 Ethernet MAC Controller Timing: MII/RMII Asynchronous Signal**

| Parameter[1, 2] | | Min | Max | Unit |
|---|---|---|---|---|
| $t_{ECOLH}$ | COL Pulse Width High | $t_{ETxCLK} \times 1.5$<br>$t_{ERxCLK} \times 1.5$ | | ns<br>ns |
| $t_{ECOLL}$ | COL Pulse Width Low | $t_{ETxCLK} \times 1.5$<br>$t_{ERxCLK} \times 1.5$ | | ns<br>ns |
| $t_{ECRSH}$ | CRS Pulse Width High | $t_{ETxCLK} \times 1.5$ | | ns |
| $t_{ECRSL}$ | CRS Pulse Width Low | $t_{ETxCLK} \times 1.5$ | | ns |

[1] MII/RMII asynchronous signals are COL, CRS. These signals are applicable in both MII and RMII modes. The asynchronous COL input is synchronized separately to both the ETxCLK and the ERxCLK, and must have a minimum pulse width high or low at least 1.5 times the period of the slower of the two clocks.
[2] The asynchronous CRS input is synchronized to the ETxCLK, and must have a minimum pulse width high or low at least 1.5 times the period of ETxCLK.

**Table 39. 10/100 Ethernet MAC Controller Timing: MII Station Management**

| Parameter[1] | | Min | Max | Unit |
|---|---|---|---|---|
| $t_{MDIOS}$ | MDIO Input Valid to MDC Rising Edge (Setup) | 10 | | ns |
| $t_{MDCIH}$ | MDC Rising Edge to MDIO Input Invalid (Hold) | 10 | | ns |
| $t_{MDCOV}$ | MDC Falling Edge to MDIO Output Valid | 25 | | ns |
| $t_{MDCOH}$ | MDC Falling Edge to MDIO Output Invalid (Hold) | –1 | | ns |

[1] MDC/MDIO is a 2-wire serial bidirectional port for controlling one or more external PHYs. MDC is an output clock whose minimum period is programmable as a multiple of the system clock SCLK. MDIO is a bidirectional data line.

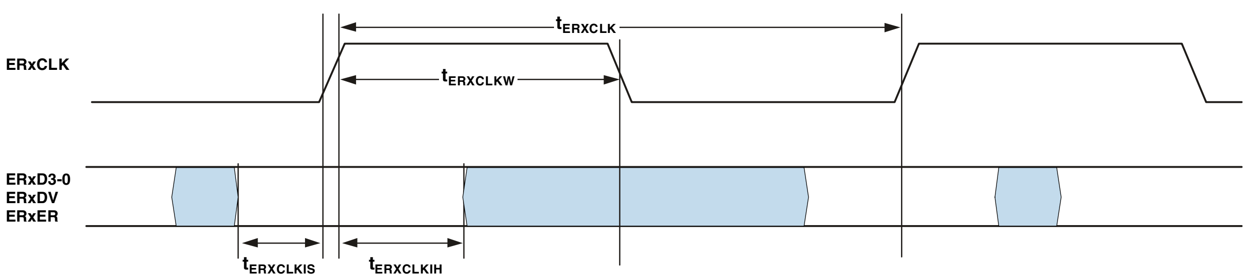



*Figure 29. 10/100 Ethernet MAC Controller Timing: MII Receive Signal*

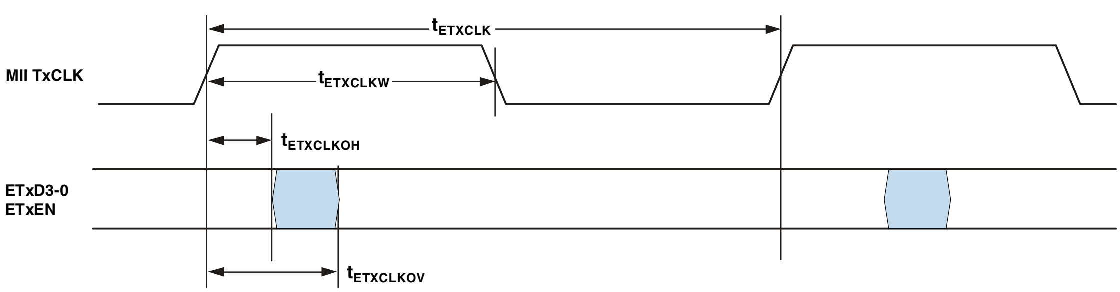



*Figure 30. 10/100 Ethernet MAC Controller Timing: MII Transmit Signal*

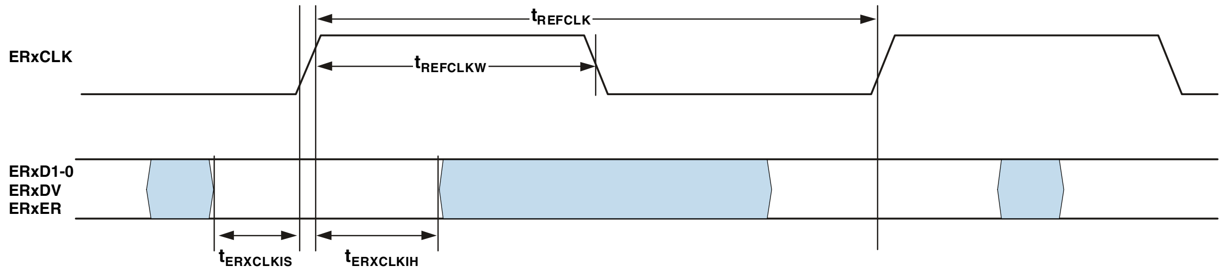



*Figure 31. 10/100 Ethernet MAC Controller Timing: RMII Receive Signal*

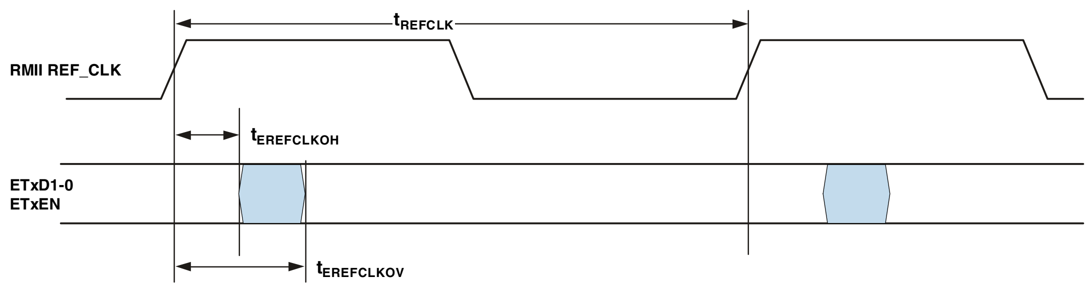



*Figure 32. 10/100 Ethernet MAC Controller Timing: RMII Transmit Signal*

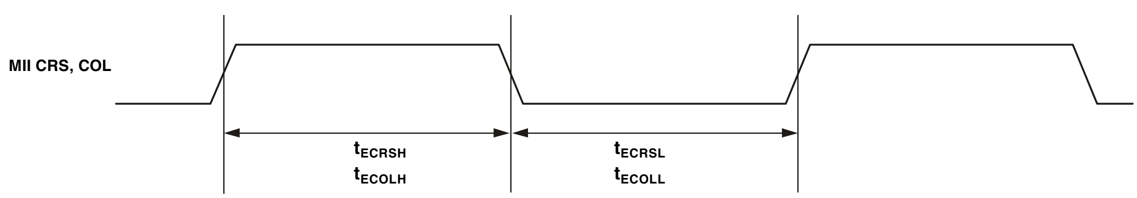



*Figure 33. 10/100 Ethernet MAC Controller Timing: Asynchronous Signal*

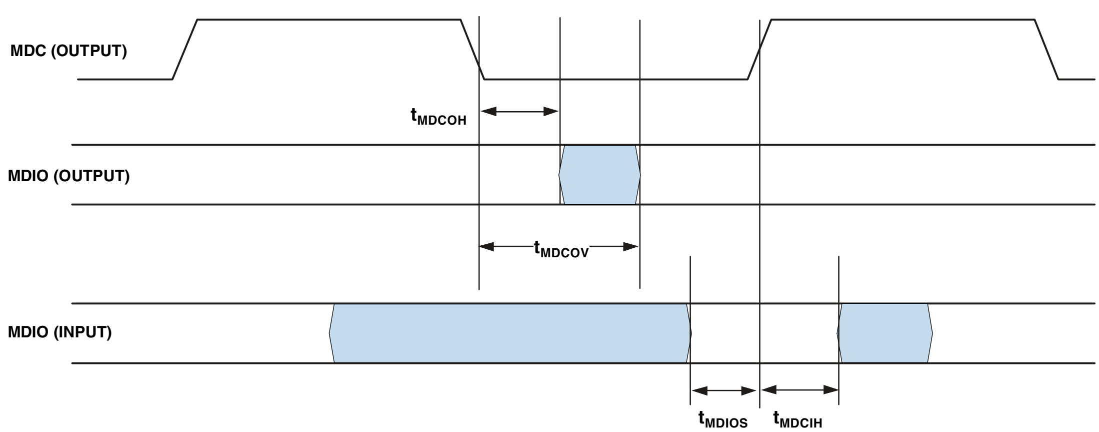



*Figure 34. 10/100 Ethernet MAC Controller Timing: MII Station Management*

## OUTPUT DRIVE CURRENTS

Figure 35 through Figure 46 show typical current-voltage characteristics for the output drivers of the processors. The curves represent the current drive capability of the output drivers as a function of output voltage. See Table 9 on Page 19 for information about which driver type corresponds to a particular pin.

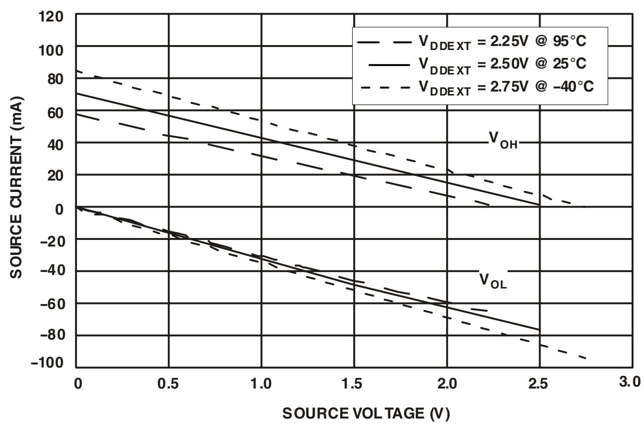



*Figure 35. Drive Current A (Low $V_{DDEXT}$)*

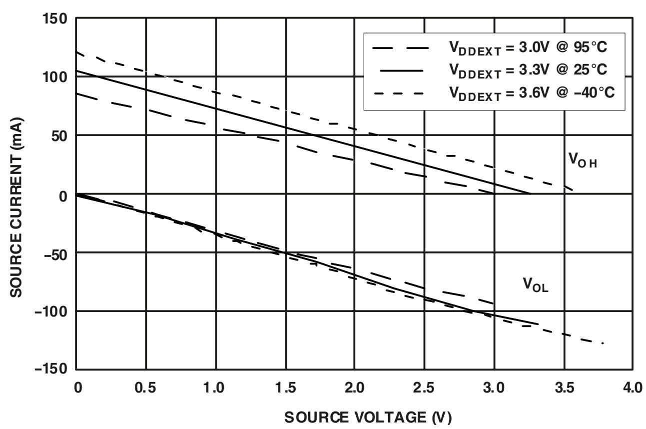



*Figure 36. Drive Current A (High $V_{DDEXT}$)*

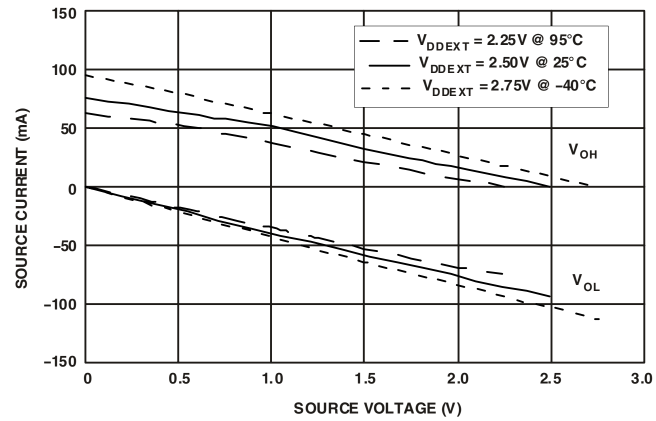



*Figure 37. Drive Current B (Low $V_{DDEXT}$)*

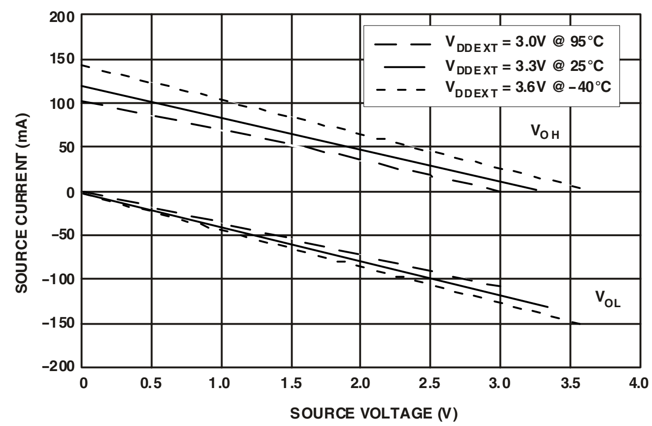



*Figure 38. Drive Current B (High $V_{DDEXT}$)*

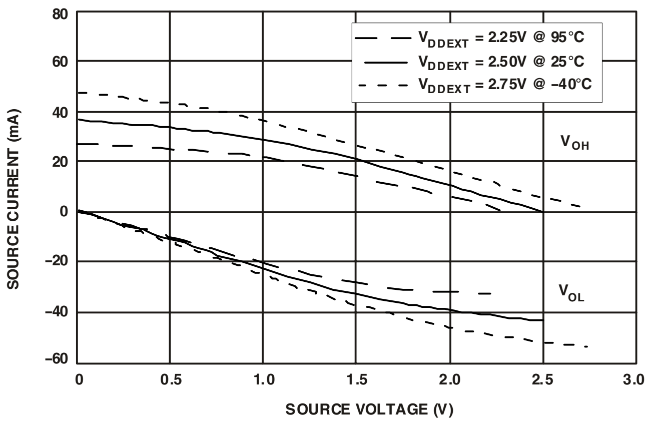



Figure 39. Drive Current C (Low $V_{DDEXT}$)

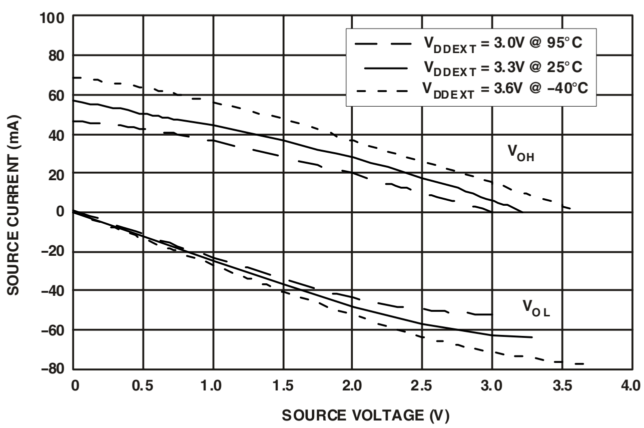



Figure 40. Drive Current C (High $V_{DDEXT}$)

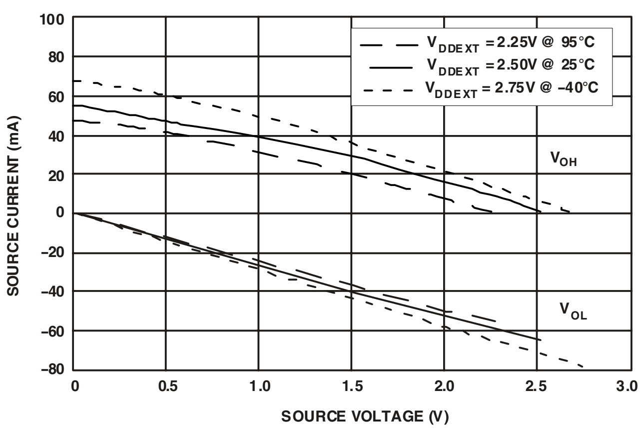



Figure 41. Drive Current D (Low $V_{DDEXT}$)

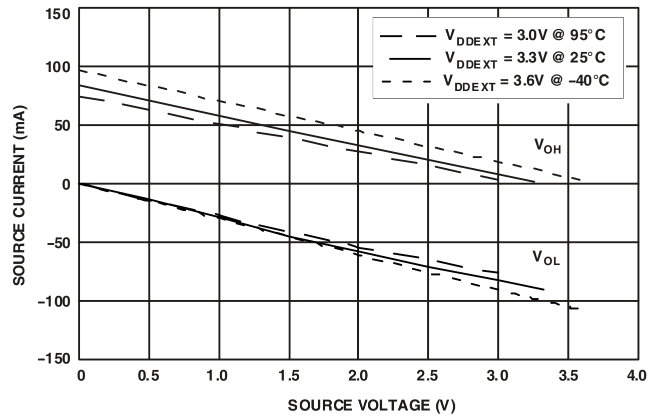



Figure 42. Drive Current D (High $V_{DDEXT}$)

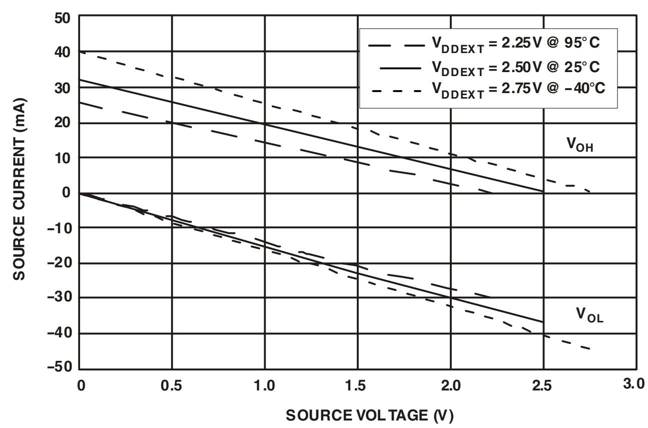



Figure 43. Drive Current E (Low $V_{DDEXT}$)

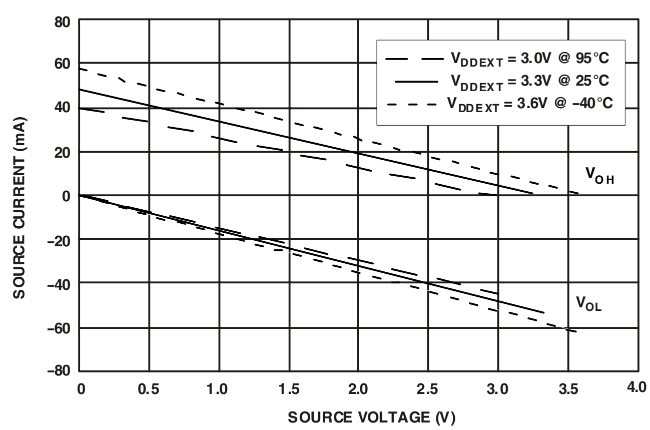



Figure 44. Drive Current E (High $V_{DDEXT}$)

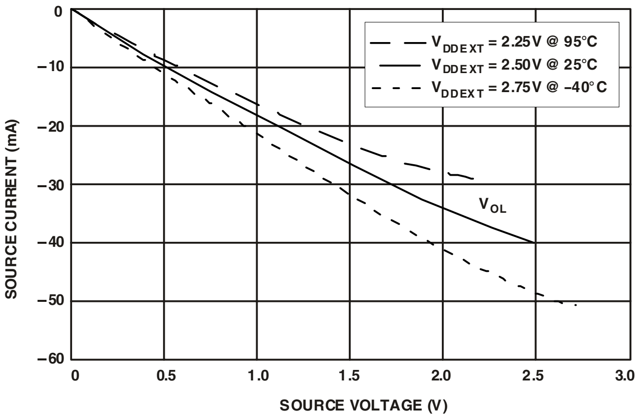



Figure 45. Drive Current F (Low $V_{DDEXT}$)

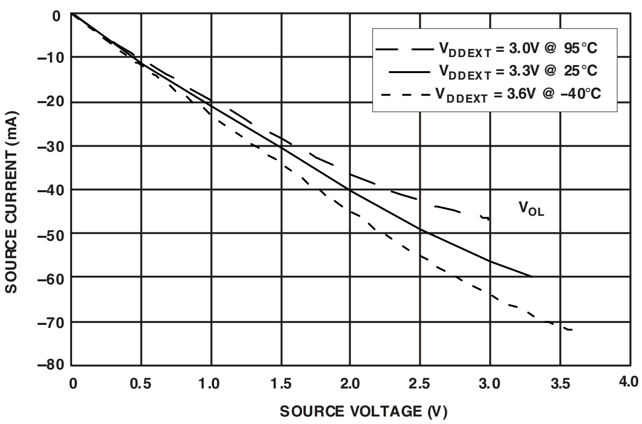



Figure 46. Drive Current F (High $V_{DDEXT}$)

## POWER DISSIPATION

Total power dissipation has two components: one due to internal circuitry ($P_{INT}$) and one due to the switching of external output drivers ($P_{EXT}$). Table 40 shows the power dissipation for internal circuitry ($V_{DDINT}$).

Many operating conditions can affect power dissipation. System designers should refer to "Estimating Power for the ADSP-BF534/BF536/BF537 Blackfin Processors" (EE-297). This document provides detailed information for optimizing designs for lowest power.

**Table 40. Internal Power Dissipation**

| Parameter | Test Conditions[1] $f_{CCLK}$ = 50 MHz $V_{DDINT}$ = 0.8 V | Test Conditions $f_{CCLK}$ = 400 MHz $V_{DDINT}$ = 1.0 V | Test Conditions $f_{CCLK}$ = 400 MHz $V_{DDINT}$ = 1.2 V | Unit |
|---|---|---|---|---|
| $I_{DDTYP}$[2] | 26 | 130 | 160 | mA |
| $I_{DDSLEEP}$[3, 4] | 16 | 30 | 37 | mA |
| $I_{DDDEEPSLEEP}$[3] | 14 | 25 | 31 | mA |
| $I_{DDHIBERNATE}$[4] | 50 | 50 | 50 | µA |
| $I_{DDRTC}$[5] | 30 | 30 | 30 | µA |
| **Parameter** | **$f_{CCLK}$ = 250 MHz $V_{DDINT}$ = 0.8 V** | **$f_{CCLK}$ = 500 MHz $V_{DDINT}$ = 1.2 V** | | **Unit** |
| $I_{DDTYP}$[2] | 65 | 190 | | mA |
| $I_{DDSLEEP}$[3, 4] | 16 | 37 | | mA |
| $I_{DDDEEPSLEEP}$[3] | 14 | 31 | | mA |
| $I_{DDHIBERNATE}$[4] | 50 | 50 | | µA |
| $I_{DDRTC}$[5] | 30 | 30 | | µA |
| **Parameter** | **$f_{CCLK}$ = 600 MHz $V_{DDINT}$ = 1.2 V** | | | **Unit** |
| $I_{DDTYP}$[2] | 220 | | | mA |
| $I_{DDSLEEP}$[3, 4] | 37 | | | mA |
| $I_{DDDEEPSLEEP}$[3] | 31 | | | mA |
| $I_{DDHIBERNATE}$[4] | 50 | | | µA |
| $I_{DDRTC}$[5] | 30 | | | µA |

[1] $I_{DD}$ data is specified for typical process parameters. All data at 25°C.
[2] Processor executing 75% dual MAC, 25% ADD with moderate data bus activity.
[3] See the *ADSP-BF537 Blackfin Processor Hardware Reference Manual* for definitions of sleep and deep sleep operating modes.
[4] $I_{DDHIBERNATE}$ is measured @ $V_{DDEXT}$ = 3.65 V with the core voltage regulator off ($V_{DDINT}$ = 0 V).
[5] Measured at $V_{DDRTC}$ = 3.3 V at 25°C.

The external component of total power dissipation is caused by the switching of output pins. Its magnitude depends on:

- The output voltage swing ($V_{DDEXT}$).
- The output capacitance ($C_0$) individual pins have to load.
- The maximum frequency ($f_0$) at which individual pins switch.

Furthermore, because I/O activity is usually not constant over time, the external component of power dissipation is not a constant value. Its peak value is best estimated by identifying representative phases with the highest I/O activity and analyzing output switching pin by pin. The following formula calculates the average power for an analyzed period by accumulating the power of all output pins.

$$\overline{P_{EXT}} = V_{DDEXT}^{\ 2} \times \sum C_0 \times \overline{f_0}$$

The frequency *f* includes driving the load high and then back low. For example: DATA15–0 pins can drive high and low at a maximum rate of $1 \div (2 \times t_{SCLK})$ while in SDRAM burst mode.

A typical power consumption can now be calculated for these conditions by adding a typical internal power dissipation:

$$P_{TOTAL} = P_{EXT} + (I_{DD} \times V_{DDINT})$$

Note that the conditions causing a worst-case $P_{EXT}$ differ from those causing a worst-case $P_{INT}$. Maximum $P_{INT}$ cannot occur while 100% of the output pins are switching from all ones (1s) to all zeros (0s). Note, as well, that it is uncommon for an application to have 100% or even 50% of the outputs switching simultaneously.

## TEST CONDITIONS

All timing parameters appearing in this data sheet were measured under the conditions described in this section.

### *Output Enable Time*

Output pins are considered to be enabled when they have made a transition from a high impedance state to the point when they start driving. The output enable time $t_{ENA}$ is the interval from the point when a reference signal reaches a high or low voltage level to the point when the output starts driving as shown in the Output Enable/Disable diagram (Figure 47). The time $t_{ENA_MEASURED}$ is the interval from when the reference signal switches to when the output voltage reaches 2.0 V (output high) or 1.0 V (output low). Time $t_{TRIP}$ is the interval from when the output starts driving to when the output reaches the 1.0 V or 2.0 V trip voltage. Time $t_{ENA}$ is calculated as shown in the equation:

$$t_{ENA} = t_{ENA_MEASURED} - t_{TRIP}$$

If multiple pins (such as the data bus) are enabled, the measurement value is that of the first pin to start driving.

### *Output Disable Time*

Output pins are considered to be disabled when they stop driving, go into a high impedance state, and start to decay from their output high or low voltage. The time for the voltage on the bus to decay by ΔV is dependent on the capacitive load, $C_L$ and the load current, $I_L$. This decay time can be approximated by the equation:

$$t_{DECAY} = (C_L \Delta V)/I_L$$

The output disable time $t_{DIS}$ is the difference between $t_{DIS_MEASURED}$ and $t_{DECAY}$ as shown in Figure 47. The time $t_{DIS_MEASURED}$ is the interval from when the reference signal switches to when the output voltage decays ΔV from the measured output-high or output-low voltage. The time $t_{DECAY}$ is calculated with test loads $C_L$ and $I_L$, and with ΔV equal to 0.5 V.

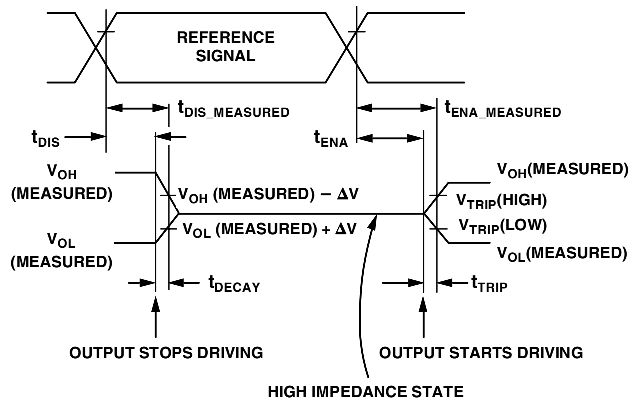



*Figure 47. Output Enable/Disable*

### *Example System Hold Time Calculation*

To determine the data output hold time in a particular system, first calculate $t_{DECAY}$ using the equation given above. Choose ΔV to be the difference between the processor's output voltage and the input threshold for the device requiring the hold time. A typical ΔV is 0.4 V. $C_L$ is the total bus capacitance (per data line), and $I_L$ is the total leakage or three-state current (per data line). The hold time is $t_{DECAY}$ plus the minimum disable time (for example, $t_{DSDAT}$ for an SDRAM write cycle).

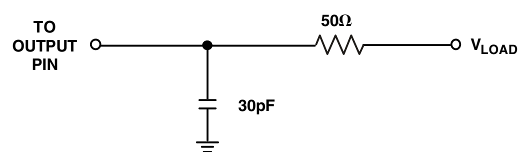



*Figure 48. Equivalent Device Loading for AC Measurements (Includes All Fixtures)*




*Figure 49. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)*

### Capacitive Loading

Output delays and holds are based on standard capacitive loads: 30 pF on all pins (see Figure 48). Figure 50 through Figure 59 on Page 58 show how output rise time varies with capacitance. The delay and hold specifications given should be derated by a factor derived from these figures. The graphs in these figures may not be linear outside the ranges shown.

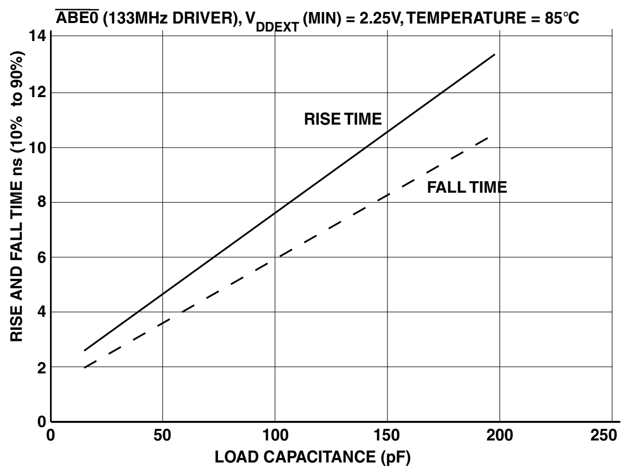



*Figure 50. Typical Output Delay or Hold for Driver A at $V_{DDEXT}$ Min*

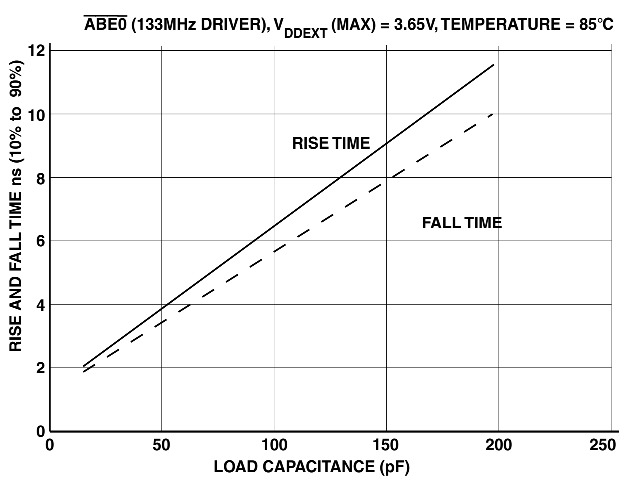



*Figure 51. Typical Output Delay or Hold for Driver A at $V_{DDEXT}$ Max*

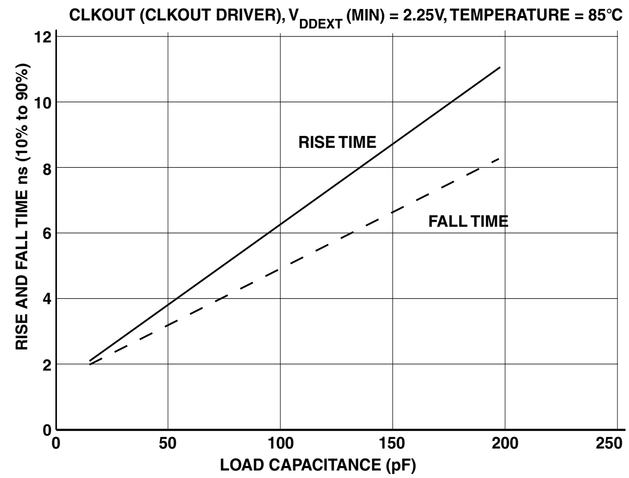



*Figure 52. Typical Output Delay or Hold for Driver B at $V_{DDEXT}$ Min*

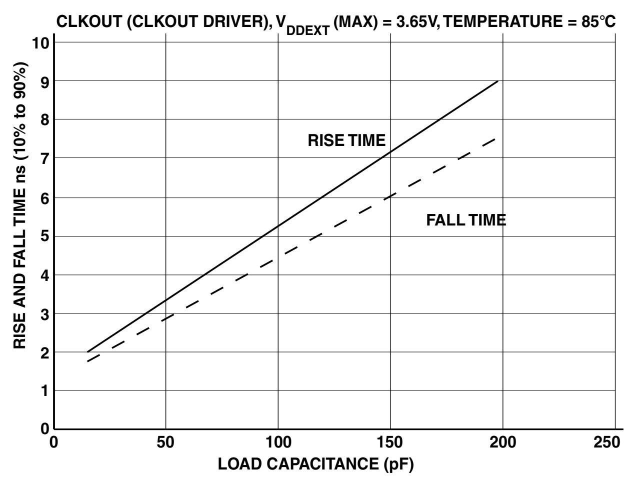



*Figure 53. Typical Output Delay or Hold for Driver B at $V_{DDEXT}$ Max*

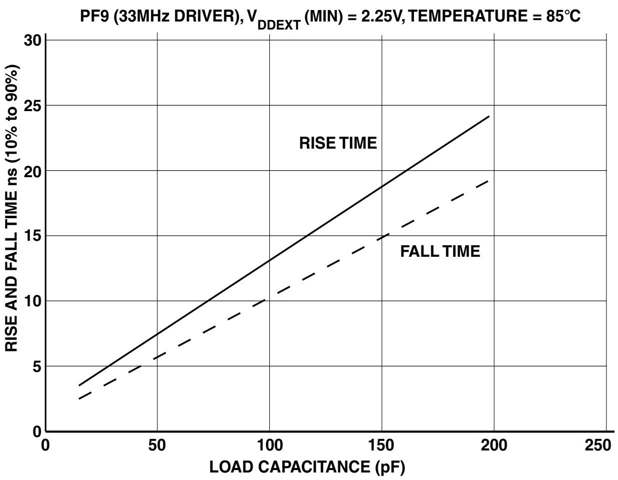



Figure 54. Typical Output Delay or Hold for Driver C at $V_{DDEXT}$ Min

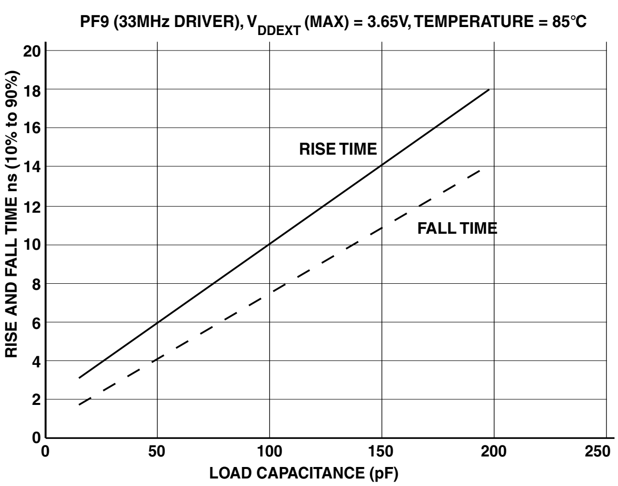



Figure 55. Typical Output Delay or Hold for Driver C at $V_{DDEXT}$ Max

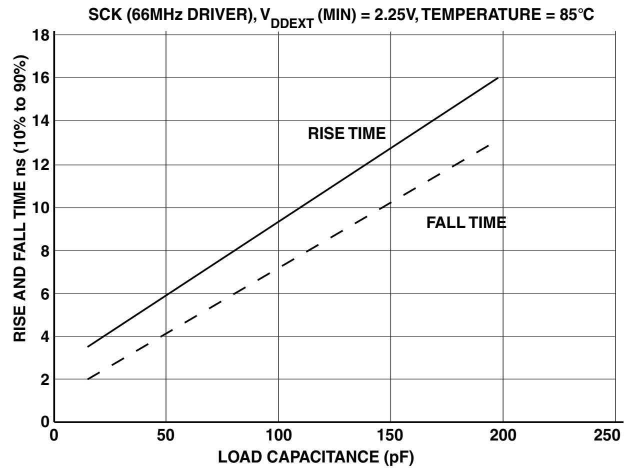



Figure 56. Typical Output Delay or Hold for Driver D at $V_{DDEXT}$ Min

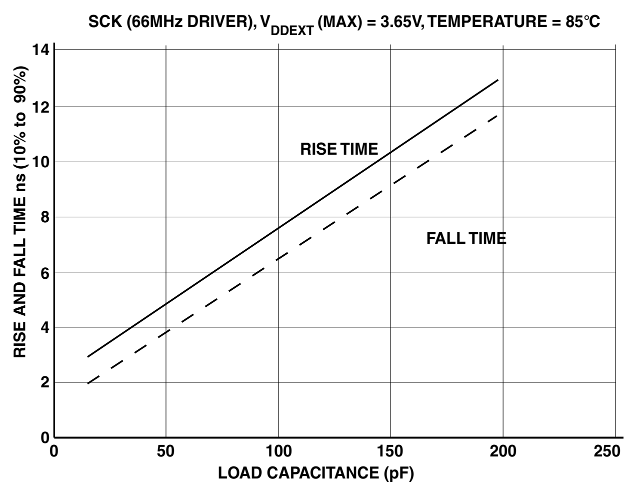



Figure 57. Typical Output Delay or Hold for Driver D at $V_{DDEXT}$ Max

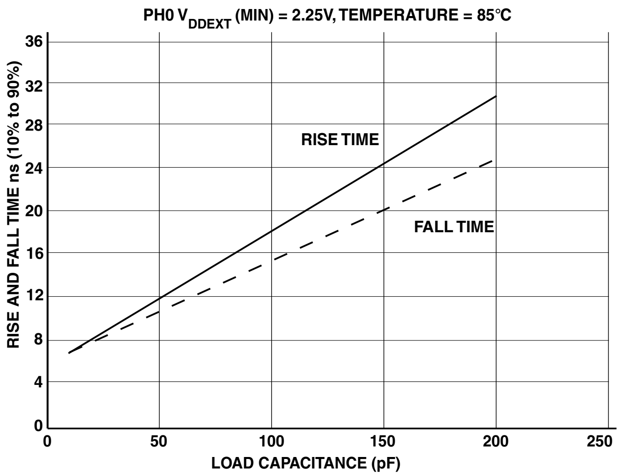



Figure 58. Typical Output Delay or Hold for Driver E at $V_{DDEXT}$ Min

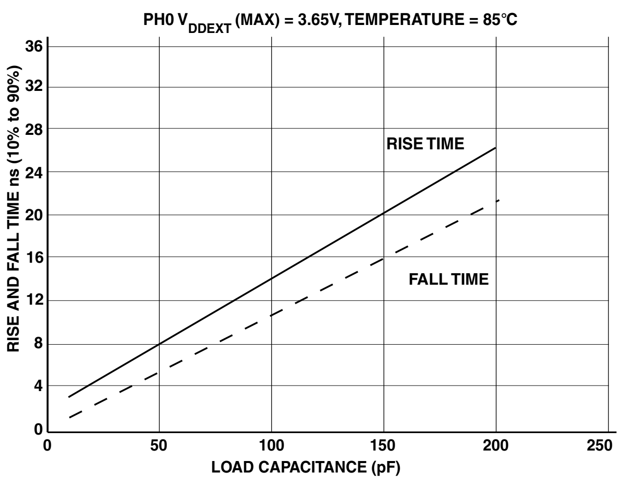



Figure 59. Typical Output Delay or Hold for Driver E at $V_{DDEXT}$ Max

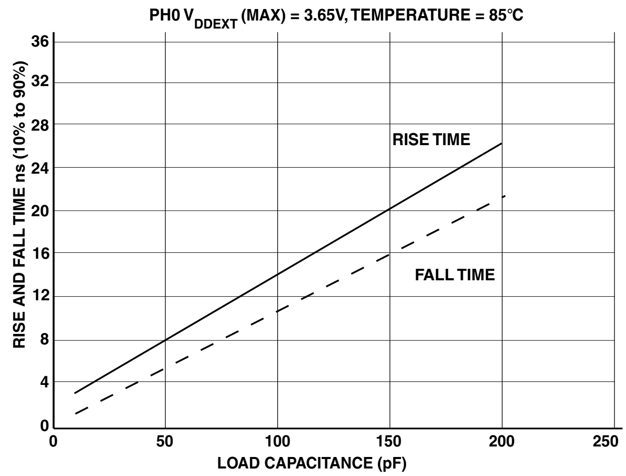



Figure 60. Typical Output Delay or Hold for Driver F at $V_{DDEXT}$ Min

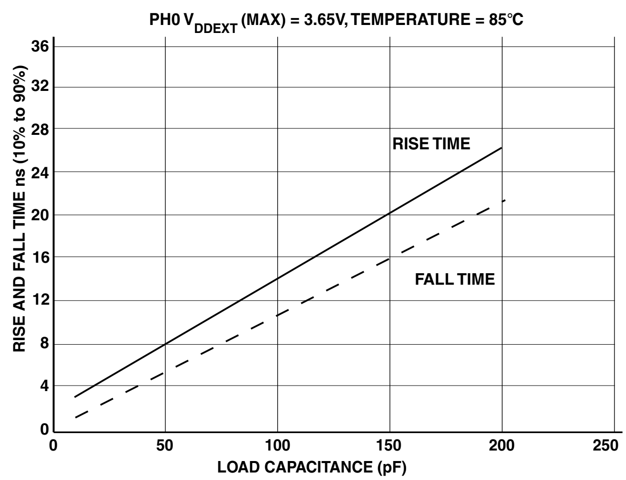



Figure 61. Typical Output Delay or Hold for Driver F at $V_{DDEXT}$ Max

## THERMAL CHARACTERISTICS

To determine the junction temperature on the application printed circuit board use:

$$T_J = T_{CASE} + (\Psi_{JT} \times P_D)$$

where:

$T_J$ = Junction temperature (°C)

$T_{CASE}$ = Case temperature (°C) measured by customer at top center of package.

$\Psi_{JT}$ = From Table 41

$P_D$ = Power dissipation (see Power Dissipation on Page 54 for the method to calculate $P_D$)

Values of $\theta_{JA}$ are provided for package comparison and printed circuit board design considerations. $\theta_{JA}$ can be used for a first order approximation of $T_J$ by the equation:

$$T_J = T_A + (\theta_{JA} \times P_D)$$

where:

$T_A$ = Ambient temperature (°C)

Values of $\theta_{JC}$ are provided for package comparison and printed circuit board design considerations when an external heat sink is required.

Values of $\theta_{JB}$ are provided for package comparison and printed circuit board design considerations.

In Table 41 through Table 43, airflow measurements comply with JEDEC standards JESD51-2 and JESD51-6, and the junction-to-board measurement complies with JESD51-8. Test board and thermal via design comply with JEDEC standards JESD51-9 (BGA). The junction-to-case measurement complies with MIL-STD-883 (Method 1012.1). All measurements use a 2S2P JEDEC test board.

Industrial applications using the 208-ball BGA package require thermal vias, to an embedded ground plane, in the PCB. Refer to JEDEC standard JESD51-9 for printed circuit board thermal ball land and thermal via design information.

**Table 41. Thermal Characteristics (182-Ball BGA)**

| Parameter | Condition | Typical | Unit |
|---|---|---|---|
| $\theta_{JA}$ | 0 linear m/s air flow | 32.80 | °C/W |
| $\theta_{JMA}$ | 1 linear m/s air flow | 29.30 | °C/W |
| $\theta_{JMA}$ | 2 linear m/s air flow | 28.00 | °C/W |
| $\theta_{JB}$ | | 20.10 | °C/W |
| $\theta_{JC}$ | | 7.92 | °C/W |
| $\Psi_{JT}$ | 0 linear m/s air flow | 0.19 | °C/W |
| $\Psi_{JT}$ | 1 linear m/s air flow | 0.35 | °C/W |
| $\Psi_{JT}$ | 2 linear m/s air flow | 0.45 | °C/W |

**Table 42. Thermal Characteristics (208-Ball BGA Without Thermal Vias in PCB)**

| Parameter | Condition | Typical | Unit |
|---|---|---|---|
| $\theta_{JA}$ | 0 linear m/s air flow | 23.30 | °C/W |
| $\theta_{JMA}$ | 1 linear m/s air flow | 20.20 | °C/W |
| $\theta_{JMA}$ | 2 linear m/s air flow | 19.20 | °C/W |
| $\theta_{JB}$ | | 13.05 | °C/W |
| $\theta_{JC}$ | | 6.92 | °C/W |
| $\Psi_{JT}$ | 0 linear m/s air flow | 0.18 | °C/W |
| $\Psi_{JT}$ | 1 linear m/s air flow | 0.27 | °C/W |
| $\Psi_{JT}$ | 2 linear m/s air flow | 0.32 | °C/W |

**Table 43. Thermal Characteristics (208-Ball BGA with Thermal Vias in PCB)**

| Parameter | Condition | Typical | Unit |
|---|---|---|---|
| $\theta_{JA}$ | 0 linear m/s air flow | 22.60 | °C/W |
| $\theta_{JMA}$ | 1 linear m/s air flow | 19.40 | °C/W |
| $\theta_{JMA}$ | 2 linear m/s air flow | 18.40 | °C/W |
| $\theta_{JB}$ | | 13.20 | °C/W |
| $\theta_{JC}$ | | 6.85 | °C/W |
| $\Psi_{JT}$ | 0 linear m/s air flow | 0.16 | °C/W |
| $\Psi_{JT}$ | 1 linear m/s air flow | 0.27 | °C/W |
| $\Psi_{JT}$ | 2 linear m/s air flow | 0.32 | °C/W |

# 182-BALL MINI-BGA PINOUT

Table 44 lists the mini-BGA pinout by signal mnemonic.
Table 45 on Page 61 lists the mini-BGA pinout by ball number.

**Table 44. 182-Ball Mini-BGA Ball Assignment (Alphabetically by Signal Mnemonic)**

| Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. |
|---|---|---|---|---|---|---|---|---|---|
| ABE0 | H13 | CLKOUT | B14 | GND | L6 | PG8 | E3 | SRAS | D13 |
| ABE1 | H12 | DATA0 | M9 | GND | L8 | PG9 | E4 | SWE | D12 |
| ADDR1 | J14 | DATA1 | N9 | GND | L10 | PH0 | C2 | TCK | P2 |
| ADDR10 | M13 | DATA10 | N6 | GND | M4 | PH1 | C3 | TDI | M3 |
| ADDR11 | M14 | DATA11 | P6 | GND | M10 | PH10 | B6 | TDO | N3 |
| ADDR12 | N14 | DATA12 | M5 | GND | P14 | PH11 | A2 | TMS | N2 |
| ADDR13 | N13 | DATA13 | N5 | NMI | B10 | PH12 | A3 | TRST | N1 |
| ADDR14 | N12 | DATA14 | P5 | PF0 | M1 | PH13 | A4 | VDDEXT | A1 |
| ADDR15 | M11 | DATA15 | P4 | PF1 | L1 | PH14 | A5 | VDDEXT | C12 |
| ADDR16 | N11 | DATA2 | P9 | PF10 | J2 | PH15 | A6 | VDDEXT | E6 |
| ADDR17 | P13 | DATA3 | M8 | PF11 | J3 | PH2 | C4 | VDDEXT | E11 |
| ADDR18 | P12 | DATA4 | N8 | PF12 | H1 | PH3 | C5 | VDDEXT | F4 |
| ADDR19 | P11 | DATA5 | P8 | PF13 | H2 | PH4 | C6 | VDDEXT | F12 |
| ADDR2 | K14 | DATA6 | M7 | PF14 | H3 | PH5 | B1 | VDDEXT | H5 |
| ADDR3 | L14 | DATA7 | N7 | PF15 | H4 | PH6 | B2 | VDDEXT | H10 |
| ADDR4 | J13 | DATA8 | P7 | PF2 | L2 | PH7 | B3 | VDDEXT | J11 |
| ADDR5 | K13 | DATA9 | M6 | PF3 | L3 | PH8 | B4 | VDDEXT | J12 |
| ADDR6 | L13 | EMU | M2 | PF4 | L4 | PH9 | B5 | VDDEXT | K7 |
| ADDR7 | K12 | GND | A10 | PF5 | K1 | PJ0 | C7 | VDDEXT | K9 |
| ADDR8 | L12 | GND | A14 | PF6 | K2 | PJ1 | B7 | VDDEXT | L7 |
| ADDR9 | M12 | GND | D4 | PF7 | K3 | PJ10 | D10 | VDDEXT | L9 |
| AMS0 | E14 | GND | E7 | PF8 | K4 | PJ11 | D11 | VDDEXT | L11 |
| AMS1 | F14 | GND | E9 | PF9 | J1 | PJ2 | B11 | VDDEXT | P1 |
| AMS2 | F13 | GND | F5 | PG0 | G1 | PJ3 | C11 | VDDINT | E5 |
| AMS3 | G12 | GND | F6 | PG1 | G2 | PJ4 | D7 | VDDINT | E8 |
| AOE | G13 | GND | F10 | PG10 | D1 | PJ5 | D8 | VDDINT | E10 |
| ARDY | E13 | GND | F11 | PG11 | D2 | PJ6 | C8 | VDDINT | G10 |
| ARE | G14 | GND | G4 | PG12 | D3 | PJ7 | B8 | VDDINT | K5 |
| AWE | H14 | GND | G5 | PG13 | D5 | PJ8 | D9 | VDDINT | K8 |
| BG | P10 | GND | G11 | PG14 | D6 | PJ9 | C9 | VDDINT | K10 |
| BGH | N10 | GND | H11 | PG15 | C1 | RESET | C10 | VDDRTC | B9 |
| BMODE0 | N4 | GND | J4 | PG2 | G3 | RTXO | A8 | VROUT0 | A13 |
| BMODE1 | P3 | GND | J5 | PG3 | F1 | RTXI | A9 | VROUT1 | B12 |
| BMODE2 | L5 | GND | J9 | PG4 | F2 | SA10 | E12 | XTAL | A11 |
| BR | D14 | GND | J10 | PG5 | F3 | SCAS | C14 | | |
| CLKBUF | A7 | GND | K6 | PG6 | E1 | SCKE | B13 | | |
| CLKIN | A12 | GND | K11 | PG7 | E2 | SMS | C13 | | |

**Table 45. 182-Ball Mini-BGA Ball Assignment (Numerically by Ball Number)**

| Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic |
|---|---|---|---|---|---|---|---|---|---|
| A1 | VDDEXT | C10 | $\overline{RESET}$ | F5 | GND | J14 | ADDR1 | M9 | DATA0 |
| A2 | PH11 | C11 | PJ3 | F6 | GND | K1 | PF5 | M10 | GND |
| A3 | PH12 | C12 | VDDEXT | F10 | GND | K2 | PF6 | M11 | ADDR15 |
| A4 | PH13 | C13 | $\overline{SMS}$ | F11 | GND | K3 | PF7 | M12 | ADDR9 |
| A5 | PH14 | C14 | $\overline{SCAS}$ | F12 | VDDEXT | K4 | PF8 | M13 | ADDR10 |
| A6 | PH15 | D1 | PG10 | F13 | $\overline{AMS2}$ | K5 | VDDINT | M14 | ADDR11 |
| A7 | CLKBUF | D2 | PG11 | F14 | $\overline{AMS1}$ | K6 | GND | N1 | $\overline{TRST}$ |
| A8 | RTXO | D3 | PG12 | G1 | PG0 | K7 | VDDEXT | N2 | TMS |
| A9 | RTXI | D4 | GND | G2 | PG1 | K8 | VDDINT | N3 | TDO |
| A10 | GND | D5 | PG13 | G3 | PG2 | K9 | VDDEXT | N4 | BMODE0 |
| A11 | XTAL | D6 | PG14 | G4 | GND | K10 | VDDINT | N5 | DATA13 |
| A12 | CLKIN | D7 | PJ4 | G5 | GND | K11 | GND | N6 | DATA10 |
| A13 | VROUT0 | D8 | PJ5 | G10 | VDDINT | K12 | ADDR7 | N7 | DATA7 |
| A14 | GND | D9 | PJ8 | G11 | GND | K13 | ADDR5 | N8 | DATA4 |
| B1 | PH5 | D10 | PJ10 | G12 | $\overline{AMS3}$ | K14 | ADDR2 | N9 | DATA1 |
| B2 | PH6 | D11 | PJ11 | G13 | $\overline{AOE}$ | L1 | PF1 | N10 | $\overline{BGH}$ |
| B3 | PH7 | D12 | $\overline{SWE}$ | G14 | $\overline{ARE}$ | L2 | PF2 | N11 | ADDR16 |
| B4 | PH8 | D13 | $\overline{SRAS}$ | H1 | PF12 | L3 | PF3 | N12 | ADDR14 |
| B5 | PH9 | D14 | $\overline{BR}$ | H2 | PF13 | L4 | PF4 | N13 | ADDR13 |
| B6 | PH10 | E1 | PG6 | H3 | PF14 | L5 | BMODE2 | N14 | ADDR12 |
| B7 | PJ1 | E2 | PG7 | H4 | PF15 | L6 | GND | P1 | VDDEXT |
| B8 | PJ7 | E3 | PG8 | H5 | VDDEXT | L7 | VDDEXT | P2 | TCK |
| B9 | VDDRTC | E4 | PG9 | H10 | VDDEXT | L8 | GND | P3 | BMODE1 |
| B10 | $\overline{NMI}$ | E5 | VDDINT | H11 | GND | L9 | VDDEXT | P4 | DATA15 |
| B11 | PJ2 | E6 | VDDEXT | H12 | $\overline{ABE1}$ | L10 | GND | P5 | DATA14 |
| B12 | VROUT1 | E7 | GND | H13 | $\overline{ABE0}$ | L11 | VDDEXT | P6 | DATA11 |
| B13 | SCKE | E8 | VDDINT | H14 | $\overline{AWE}$ | L12 | ADDR8 | P7 | DATA8 |
| B14 | CLKOUT | E9 | GND | J1 | PF9 | L13 | ADDR6 | P8 | DATA5 |
| C1 | PG15 | E10 | VDDINT | J2 | PF10 | L14 | ADDR3 | P9 | DATA2 |
| C2 | PH0 | E11 | VDDEXT | J3 | PF11 | M1 | PF0 | P10 | $\overline{BG}$ |
| C3 | PH1 | E12 | SA10 | J4 | GND | M2 | $\overline{EMU}$ | P11 | ADDR19 |
| C4 | PH2 | E13 | ARDY | J5 | GND | M3 | TDI | P12 | ADDR18 |
| C5 | PH3 | E14 | $\overline{AMS0}$ | J9 | GND | M4 | GND | P13 | ADDR17 |
| C6 | PH4 | F1 | PG3 | J10 | GND | M5 | DATA12 | P14 | GND |
| C7 | PJ0 | F2 | PG4 | J11 | VDDEXT | M6 | DATA9 | | |
| C8 | PJ6 | F3 | PG5 | J12 | VDDEXT | M7 | DATA6 | | |
| C9 | PJ9 | F4 | VDDEXT | J13 | ADDR4 | M8 | DATA3 | | |

Figure 63 shows the top view of the mini-BGA ball configuration. Figure 62 shows the bottom view of the mini-BGA ball configuration.

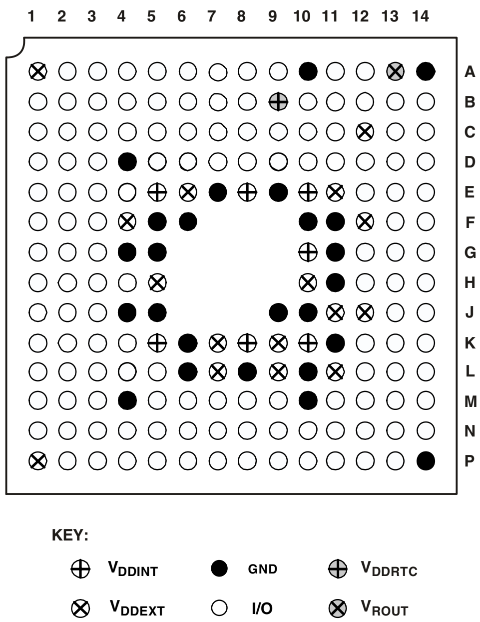



*Figure 62. 182-Ball Mini-BGA Configuration (Top View)*

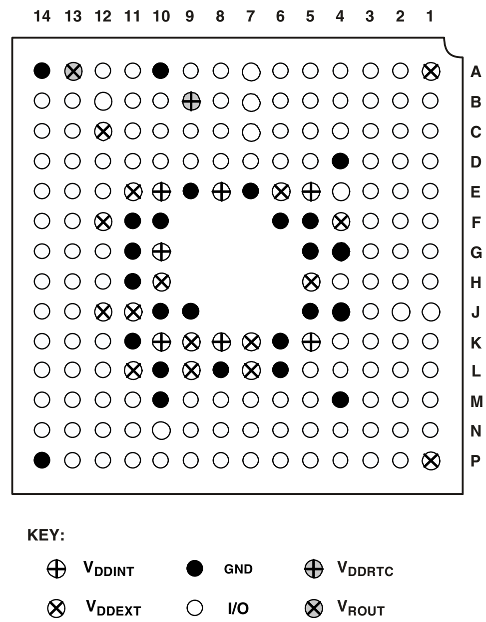



*Figure 63. 182-Ball Mini-BGA Configuration (Bottom View)*

# 208-BALL SPARSE MINI-BGA PINOUT

Table 46 lists the sparse mini-BGA pinout by signal mnemonic. Table 47 on Page 64 lists the sparse mini-BGA pinout by ball number.

**Table 46. 208-Ball Sparse Mini-BGA Ball Assignment (Alphabetically by Signal Mnemonic)**

| Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. |
|---|---|---|---|---|---|---|---|---|---|
| $\overline{\text{ABE0}}$ | P19 | DATA12 | Y4 | GND | M13 | PG6 | E2 | TDI | V1 |
| $\overline{\text{ABE1}}$ | P20 | DATA13 | W4 | GND | N9 | PG7 | D1 | TDO | Y2 |
| ADDR1 | R19 | DATA14 | Y3 | GND | N10 | PG8 | D2 | TMS | U2 |
| ADDR10 | W18 | DATA15 | W3 | GND | N11 | PG9 | C1 | $\overline{\text{TRST}}$ | U1 |
| ADDR11 | Y18 | DATA2 | Y9 | GND | N12 | PH0 | B4 | VDDEXT | G7 |
| ADDR12 | W17 | DATA3 | W9 | GND | N13 | PH1 | A5 | VDDEXT | G8 |
| ADDR13 | Y17 | DATA4 | Y8 | GND | P11 | PH10 | B9 | VDDEXT | G9 |
| ADDR14 | W16 | DATA5 | W8 | GND | V2 | PH11 | A10 | VDDEXT | G10 |
| ADDR15 | Y16 | DATA6 | Y7 | GND | W2 | PH12 | B10 | VDDEXT | H7 |
| ADDR16 | W15 | DATA7 | W7 | GND | W19 | PH13 | A11 | VDDEXT | H8 |
| ADDR17 | Y15 | DATA8 | Y6 | GND | Y1 | PH14 | B11 | VDDEXT | J7 |
| ADDR18 | W14 | DATA9 | W6 | GND | Y13 | PH15 | A12 | VDDEXT | J8 |
| ADDR19 | Y14 | $\overline{\text{EMU}}$ | T1 | GND | Y20 | PH2 | B5 | VDDEXT | K7 |
| ADDR2 | T20 | GND | A1 | $\overline{\text{NMI}}$ | C20 | PH3 | A6 | VDDEXT | K8 |
| ADDR3 | T19 | GND | A13 | PF0 | T2 | PH4 | B6 | VDDEXT | L7 |
| ADDR4 | U20 | GND | A20 | PF1 | R1 | PH5 | A7 | VDDEXT | L8 |
| ADDR5 | U19 | GND | B2 | PF10 | L2 | PH6 | B7 | VDDEXT | M7 |
| ADDR6 | V20 | GND | G11 | PF11 | K1 | PH7 | A8 | VDDEXT | M8 |
| ADDR7 | V19 | GND | H9 | PF12 | K2 | PH8 | B8 | VDDEXT | N7 |
| ADDR8 | W20 | GND | H10 | PF13 | J1 | PH9 | A9 | VDDEXT | N8 |
| ADDR9 | Y19 | GND | H11 | PF14 | J2 | PJ0 | B12 | VDDEXT | P7 |
| $\overline{\text{AMS0}}$ | M20 | GND | H12 | PF15 | H1 | PJ1 | B13 | VDDEXT | P8 |
| $\overline{\text{AMS1}}$ | M19 | GND | H13 | PF2 | R2 | PJ10 | B19 | VDDEXT | P9 |
| $\overline{\text{AMS2}}$ | G20 | GND | J9 | PF3 | P1 | PJ11 | C19 | VDDEXT | P10 |
| $\overline{\text{AMS3}}$ | G19 | GND | J10 | PF4 | P2 | PJ2 | D19 | VDDINT | G12 |
| $\overline{\text{AOE}}$ | N20 | GND | J11 | PF5 | N1 | PJ3 | E19 | VDDINT | G13 |
| ARDY | J19 | GND | J12 | PF6 | N2 | PJ4 | B18 | VDDINT | G14 |
| $\overline{\text{ARE}}$ | N19 | GND | J13 | PF7 | M1 | PJ5 | A19 | VDDINT | H14 |
| $\overline{\text{AWE}}$ | R20 | GND | K9 | PF8 | M2 | PJ6 | B15 | VDDINT | J14 |
| $\overline{\text{BG}}$ | Y11 | GND | K10 | PF9 | L1 | PJ7 | B16 | VDDINT | K14 |
| $\overline{\text{BGH}}$ | Y12 | GND | K11 | PG0 | H2 | PJ8 | B17 | VDDINT | L14 |
| BMODE0 | W13 | GND | K12 | PG1 | G1 | PJ9 | B20 | VDDINT | M14 |
| BMODE1 | W12 | GND | K13 | PG10 | C2 | $\overline{\text{RESET}}$ | D20 | VDDINT | N14 |
| BMODE2 | W11 | GND | L9 | PG11 | B1 | RTXO | A15 | VDDINT | P12 |
| $\overline{\text{BR}}$ | F19 | GND | L10 | PG12 | A2 | RTXI | A14 | VDDINT | P13 |
| CLKBUF | B14 | GND | L11 | PG13 | A3 | SA10 | L20 | VDDINT | P14 |
| CLKIN | A18 | GND | L12 | PG14 | B3 | $\overline{\text{SCAS}}$ | K20 | VDDRTC | A16 |
| CLKOUT | H19 | GND | L13 | PG15 | A4 | SCKE | H20 | VROUT0 | E20 |
| DATA0 | Y10 | GND | M9 | PG2 | G2 | $\overline{\text{SMS}}$ | J20 | VROUT1 | F20 |
| DATA1 | W10 | GND | M10 | PG3 | F1 | $\overline{\text{SRAS}}$ | K19 | XTAL | A17 |
| DATA10 | Y5 | GND | M11 | PG4 | F2 | $\overline{\text{SWE}}$ | L19 | | |
| DATA11 | W5 | GND | M12 | PG5 | E1 | TCK | W1 | | |

Table 47 lists the sparse mini-BGA pinout by ball number.
Table 46 on Page 63 lists the sparse mini-BGA pinout by signal mnemonic.

**Table 47. 208-Ball Sparse Mini-BGA Ball Assignment (Numerically by Ball Number)**

| Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic | Ball No. | Mnemonic |
|---|---|---|---|---|---|---|---|---|---|
| A1 | GND | C19 | PJ11 | J9 | GND | M19 | AMS1 | W1 | TCK |
| A2 | PG12 | C20 | NMI | J10 | GND | M20 | AMS0 | W2 | GND |
| A3 | PG13 | D1 | PG7 | J11 | GND | N1 | PF5 | W3 | DATA15 |
| A4 | PG15 | D2 | PG8 | J12 | GND | N2 | PF6 | W4 | DATA13 |
| A5 | PH1 | D19 | PJ2 | J13 | GND | N7 | VDDEXT | W5 | DATA11 |
| A6 | PH3 | D20 | RESET | J14 | VDDINT | N8 | VDDEXT | W6 | DATA9 |
| A7 | PH5 | E1 | PG5 | J19 | ARDY | N9 | GND | W7 | DATA7 |
| A8 | PH7 | E2 | PG6 | J20 | SMS | N10 | GND | W8 | DATA5 |
| A9 | PH9 | E19 | PJ3 | K1 | PF11 | N11 | GND | W9 | DATA3 |
| A10 | PH11 | E20 | VROUT0 | K2 | PF12 | N12 | GND | W10 | DATA1 |
| A11 | PH13 | F1 | PG3 | K7 | VDDEXT | N13 | GND | W11 | BMODE2 |
| A12 | PH15 | F2 | PG4 | K8 | VDDEXT | N14 | VDDINT | W12 | BMODE1 |
| A13 | GND | F19 | BR | K9 | GND | N19 | ARE | W13 | BMODE0 |
| A14 | RTXI | F20 | VROUT1 | K10 | GND | N20 | AOE | W14 | ADDR18 |
| A15 | RTXO | G1 | PG1 | K11 | GND | P1 | PF3 | W15 | ADDR16 |
| A16 | VDDRTC | G2 | PG2 | K12 | GND | P2 | PF4 | W16 | ADDR14 |
| A17 | XTAL | G7 | VDDEXT | K13 | GND | P7 | VDDEXT | W17 | ADDR12 |
| A18 | CLKIN | G8 | VDDEXT | K14 | VDDINT | P8 | VDDEXT | W18 | ADDR10 |
| A19 | PJ5 | G9 | VDDEXT | K19 | SRAS | P9 | VDDEXT | W19 | GND |
| A20 | GND | G10 | VDDEXT | K20 | SCAS | P10 | VDDEXT | W20 | ADDR8 |
| B1 | PG11 | G11 | GND | L1 | PF9 | P11 | GND | Y1 | GND |
| B2 | GND | G12 | VDDINT | L2 | PF10 | P12 | VDDINT | Y2 | TDO |
| B3 | PG14 | G13 | VDDINT | L7 | VDDEXT | P13 | VDDINT | Y3 | DATA14 |
| B4 | PH0 | G14 | VDDINT | L8 | VDDEXT | P14 | VDDINT | Y4 | DATA12 |
| B5 | PH2 | G19 | AMS3 | L9 | GND | P19 | ABE0 | Y5 | DATA10 |
| B6 | PH4 | G20 | AMS2 | L10 | GND | P20 | ABE1 | Y6 | DATA8 |
| B7 | PH6 | H1 | PF15 | L11 | GND | R1 | PF1 | Y7 | DATA6 |
| B8 | PH8 | H2 | PG0 | L12 | GND | R2 | PF2 | Y8 | DATA4 |
| B9 | PH10 | H7 | VDDEXT | L13 | GND | R19 | ADDR1 | Y9 | DATA2 |
| B10 | PH12 | H8 | VDDEXT | L14 | VDDINT | R20 | AWE | Y10 | DATA0 |
| B11 | PH14 | H9 | GND | L19 | SWE | T1 | EMU | Y11 | BG |
| B12 | PJ0 | H10 | GND | L20 | SA10 | T2 | PF0 | Y12 | BGH |
| B13 | PJ1 | H11 | GND | M1 | PF7 | T19 | ADDR3 | Y13 | GND |
| B14 | CLKBUF | H12 | GND | M2 | PF8 | T20 | ADDR2 | Y14 | ADDR19 |
| B15 | PJ6 | H13 | GND | M7 | VDDEXT | U1 | TRST | Y15 | ADDR17 |
| B16 | PJ7 | H14 | VDDINT | M8 | VDDEXT | U2 | TMS | Y16 | ADDR15 |
| B17 | PJ8 | H19 | CLKOUT | M9 | GND | U19 | ADDR5 | Y17 | ADDR13 |
| B18 | PJ4 | H20 | SCKE | M10 | GND | U20 | ADDR4 | Y18 | ADDR11 |
| B19 | PJ10 | J1 | PF13 | M11 | GND | V1 | TDI | Y19 | ADDR9 |
| B20 | PJ9 | J2 | PF14 | M12 | GND | V2 | GND | Y20 | GND |
| C1 | PG9 | J7 | VDDEXT | M13 | GND | V19 | ADDR7 | | |
| C2 | PG10 | J8 | VDDEXT | M14 | VDDINT | V20 | ADDR6 | | |

Figure 64 shows the top view of the sparse mini-BGA ball configuration. Figure 65 shows the bottom view of the sparse mini-BGA ball configuration.

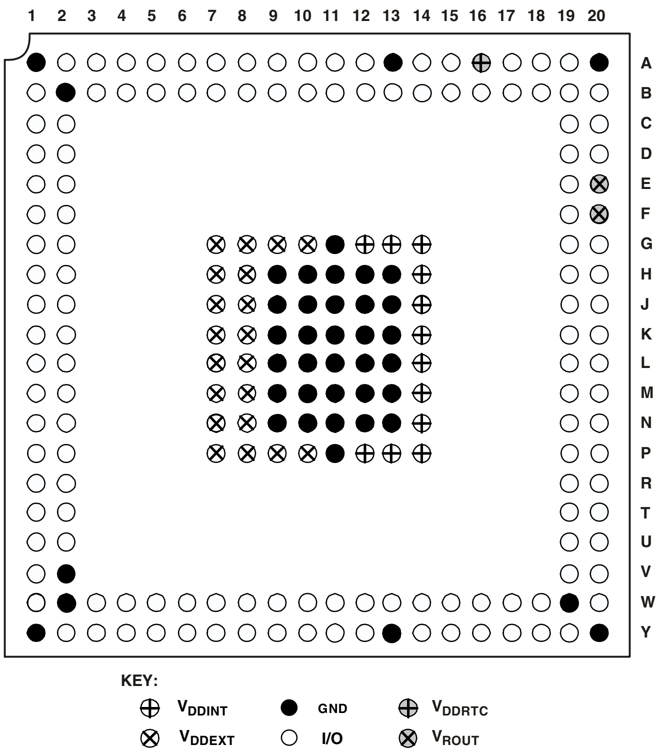



*Figure 64. 208-Ball Mini-BGA Configuration (Top View)*

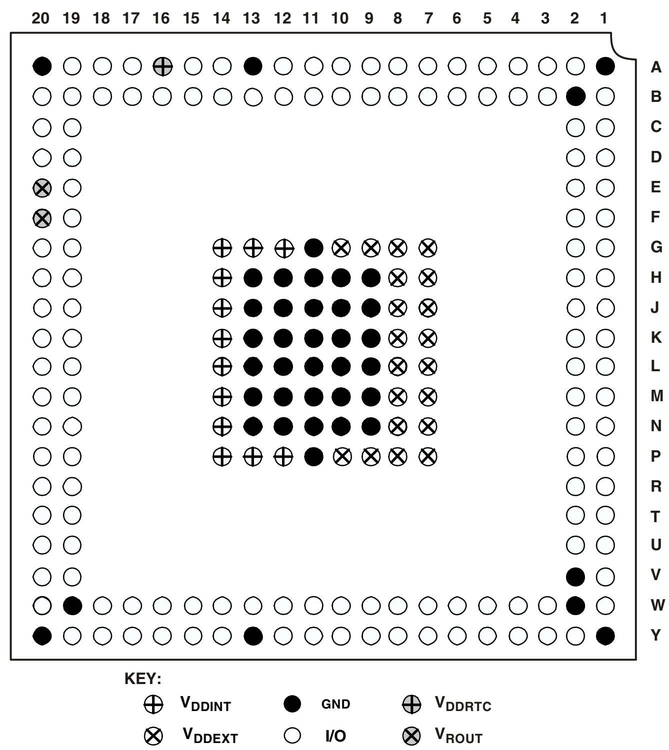



*Figure 65. 208-Ball Mini-BGA Configuration (Bottom View)*

# OUTLINE DIMENSIONS

Dimensions in Figure 66 and Figure 67 are shown in millimeters.

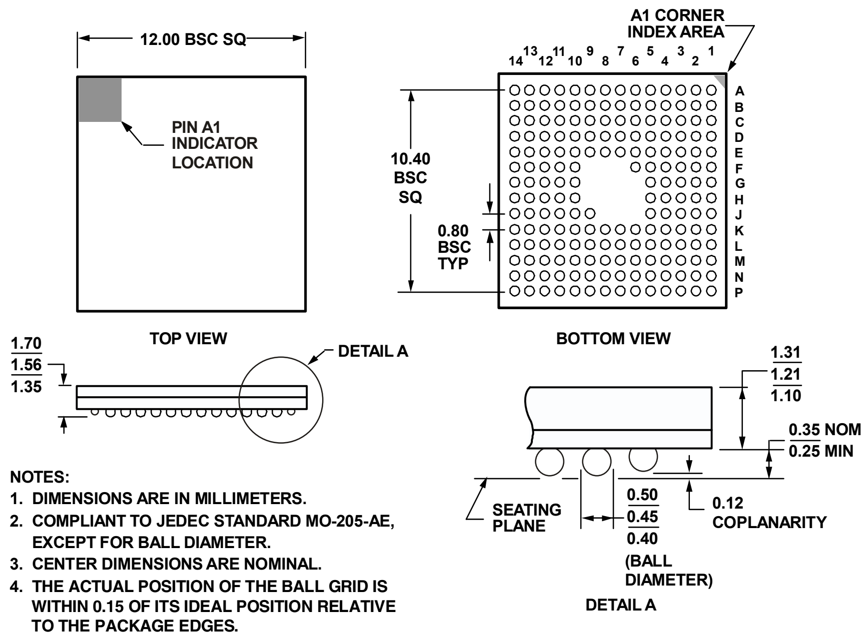



*Figure 66. 182-Ball Mini-BGA (BC-182)*

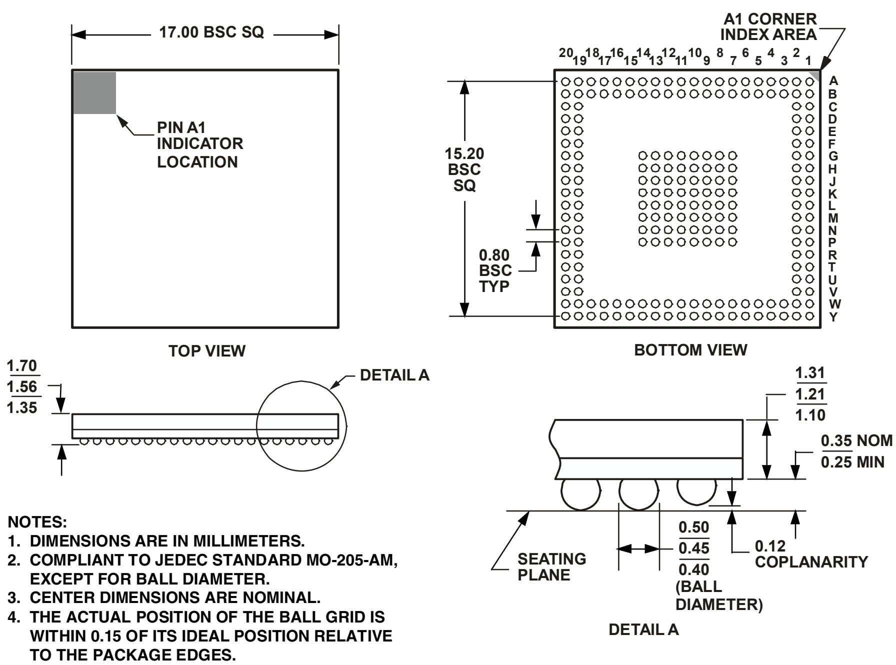



*Figure 67. 208-Ball Sparse Mini-BGA (BC-208-2)*

## SURFACE MOUNT DESIGN

The following table is provided as an aide to PCB design. For industry-standard design recommendations, refer to IPC-7351, *Generic Requirements for Surface Mount Design and Land Pattern Standard*.

| Package | Ball Attach Type | Solder Mask Opening | Ball Pad Size |
|---|---|---|---|
| 182-Ball Mini-BGA (BC-182) | Solder Mask Defined | 0.40 mm diameter | 0.55 mm diameter |
| 208-Ball Sparse Mini-BGA (BC-208-2) | Solder Mask Defined | 0.40 mm diameter | 0.55 mm diameter |

## ORDERING GUIDE

| Model | Temperature Range[1] | Speed Grade (Max) | Operating Voltage (Nominal) | Package Description | Package Option |
|---|---|---|---|---|---|
| ADSP-BF534BBC-4A | –40°C to +85°C | 400 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF534BBCZ-4A[2] | –40°C to +85°C | 400 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF534BBC-5A | –40°C to +85°C | 500 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF534BBCZ-5A[2] | –40°C to +85°C | 500 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF534BBCZ-4B[2] | –40°C to +85°C | 400 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |
| ADSP-BF534BBCZ-5B[2] | –40°C to +85°C | 500 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |
| ADSP-BF534WYBCZ-4B[2, 3] | –40°C to +105°C | 400 MHz | 1.2 V internal, 3.0 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |
| ADSP-BF534WBBCZ-4A[2, 3] | –40°C to +85°C | 400 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF534WBBCZ-4B[2, 3] | –40°C to +85°C | 400 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |
| ADSP-BF534WBBCZ-5B[2, 3] | –40°C to +85°C | 500 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |
| ADSP-BF536BBC-3A | –40°C to +85°C | 300 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF536BBCZ-3A[2] | –40°C to +85°C | 300 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF536BBC-4A | –40°C to +85°C | 400 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF536BBCZ-4A[2] | –40°C to +85°C | 400 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF536BBCZ-3B[2] | –40°C to +85°C | 300 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |
| ADSP-BF536BBCZ-4B[2] | –40°C to +85°C | 400 MHz | 1.2 V internal, 2.5 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |
| ADSP-BF537BBC-5A | –40°C to +85°C | 500 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF537BBCZ-5A[2] | –40°C to +85°C | 500 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF537KBC-6A | 0°C to 70°C | 600 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF537KBCZ-6A[2] | 0°C to 70°C | 600 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 182-Ball Mini-BGA | BC-182 |
| ADSP-BF537BBCZ-5B[2] | –40°C to +85°C | 500 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |
| ADSP-BF537KBCZ-6B[2] | 0°C to 70°C | 600 MHz | 1.26 V internal, 2.5 V or 3.3 V I/O | 208-Ball Sparse Mini-BGA | BC-208-2 |

[1] Referenced temperature is ambient temperature.
[2] Z = Pb-free part.
[3] The W in the model number signifies that a version of this product is available for use in automotive applications. Contact your local ADI sales office for complete ordering information.